AIAA Aviation
22-26 June 2015, Dallas, TX
22nd AIAA Computational Fluid Dynamics Conference

AIAA 2015-3051

Towards high-fidelity aerospace design in the age of extreme scale supercomputing

Qiqi Wang*

This paper describes the prospect of high-fidelity simulation and design. We argue that high-fidelity design must be significantly faster, preferably real-time, for it to reach its full potential. This paper describes two relatively new research directions that can contribute to this goal.

I. Introduction

Aerospace engineers have always been leading in the field of design using computational simulations. We design vehicles that go fast, fly high, and operate in extreme conditions. Because computation can help reduce expensive tests that physically simulate these extreme conditions, we benefit from computational simulation and design. This is why successful aerospace companies and agencies, such as NASA, have always been leading and investing in research and development of computational tools for simulation and design.[1,2]

Aerospace engineers have been leading contributors to computational tools and technology. NASA, for example, led the ICASE program which produced significant advance Computational Fluid Dynamics (CFD) methods.[3] Advance in CFD revolutionized many aspects of aerospace engineering, including how commercial jets are designed. It helped reduce the cost of testing wing designs in wind tunnels when designing new airplanes. For example, Boeing had to test 77 wing designs when designing the 757 in late 1970s. By mid 1990s, it only needed to test 11 wing designs when designing the 737-NG.[4] Our investment in computational simulation and design, including CFD, has paid back.

Despite decades of progress, computational simulation and design, including CFD, is far from reaching its limit. High fidelity CFD simulations, in particular, may change the world of aerospace engineering in the near future. In 2014, NASA completed the CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences.[2] The Vision 2030 study highlighted the enormous potential of simulations and design based on high-fidelity CFD. High fidelity CFD includes Large Eddy Simulations (LES), wall-modeled LES, and hybrid RANS-LES. These unsteady simulations can capture more aerospace-relevant flow physics, work for complex geometries, and are becoming increasingly reasonable in terms of computational cost. Some of those leading the CFD Vision 2030 study predict that aerospace engineers will be using these high-fidelity simulations in the design, almost in real time.[5]

Computational simulations would be most useful if they are reliable enough to be trusted, and fast enough to be applied in real-time design. If this is achieved, it might become as easy to design and test aerospace components virtually as to design and test as bicycle components. They would not only accelerate aerospace design and development cycles, reduce the cost, but also may spawn a culture of unprecedented innovation in aerospace engineering.

II. How supercomputing can enable high-fidelity design

Peta-scale supercomputing is already enabling high-fidelity design. NASA, for example, succeeded in using high-fidelity simulation to assess the aeroacoustic load of the Space Launch System, a rocket designed

*Associate Professor, Aeronautics and Astronautics, Cambridge, MA, USA, AIAA member

to replace the Space Shuttle.[6] The simulation matched wind tunnel measurements and gave engineers much insight into the cause of the vibrations and how to design the rocket to reduce the vibration. NASA also demonstrated adjoint-based design capability for turbulent flow over a UH60 helicopter in forward flight.[7] High-fidelity design is happening, right now.

The real question, however, is whether high-fidelity design will be widely adopted in aerospace engineering. Will it become a common and widely-available tool? Will it revolutionize how aerospace engineers work? Will such tools lead to vastly accelerated and cheaper aerospace design cycles? Will they enable a culture of unprecedented innovation?

The answer to these questions hinges on whether high fidelity design tools can become not only higher fidelity, but also significantly faster and cheaper. Design using high fidelity simulations are currently very slow. Even on a high-end supercomputer, each simulation typically runs for days or weeks.[2,7] This is mainly because high-fidelity CFD simulations often require sequentially advancing through tens of thousands to millions of time steps. These unsteady simulations are parallelized only in the spatial domain, because efforts towards parallelizing them in time is challenged by complex nonlinear dynamics these simulations can exhibit.[8,9] In addition, simulation-based design usually takes many iterations of such simulations, and therefore takes at least months to complete. Methods to accelerate CFD-based design, such as the adjoint method, is scarcely available, and is also challenged by complex nonlinear dynamics of these simulations.[10] As a result, high-fidelity design is attempted only by the most visionary, brave and resourceful pioneers in aerospace engineering. To become a slide rule for aerospace engineers, high-fidelity design must be faster, orders of magnitude faster.

I believe we can and will make high fidelity design orders of magnitude faster. To do so, we need to transform current software tools and technologies used in simulation and design. These tools and technologies must not only significantly accelerate high-fidelity simulations, but also significantly reduce the number of iterations needed to complete a high-fidelity design. These two goals are pursued by researchers in many fields. Better numerical algorithms, e.g., high order methods,[11,12] aims to significantly reduce the degrees of freedom needed to perform accurate CFD simulations, thereby accelerating it. Better treatment of turbulence, e.g., wall models for Large Eddy Simulations, can also significantly accelerate these simulation.[13–15] Parallel-in-time methods can scale a moderate-sized high-fidelity simulation to very large parallel computers, leading to faster simulations.[8] Better optimization algorithms,[16] including those that can exploit additional level of parallism,[17] can lead to faster design optimization. Acceleration achieved in these fields, together with improvements in computing hardware, have multiplicative effect. If considerable progress is made in each of these fields, combining them would lead to significant acceleration to high-fidelity design.

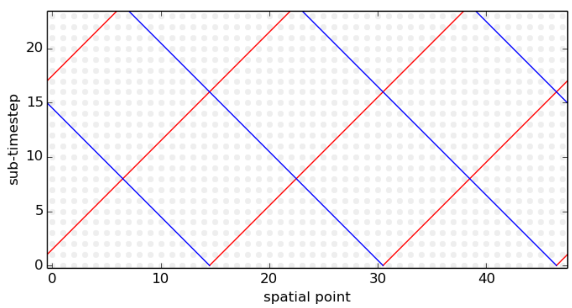
In addition, new ways of accelerating high-fidelity design is important to explore, precisely because of this multiplicative effect. The rest of this paper describes two new and promising research directions towards these two goals. Section 3 is devoted to the swept decomposition rule, a new way to accelerate high-fidelity simulations; Section 4 is devoted to for high-fidelity design.

III. Towards faster high-fidelity simulations: Swept domain decomposition rule for breaking the latency barrier

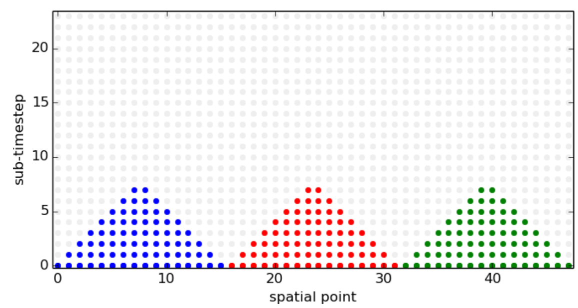
CFD is currently limited by the latency barrier. In a parallel simulation, computing nodes communicate to each other frequently. Each communication takes at least a few microseconds, and on a common network, tens of microseconds. This minimum communication time, regardless of how much information is communicated, is the network latency.[18,19] If network latency exceeds the computing time between consecutive communications, scaling to more nodes does not accelerate the simulation.[18] This barrier to scaling is called the latency barrier. Network latency improves slowly, on average by no more than 10 to 20 percent per year.[20] To significantly accelerate high-fidelity CFD, the latency barrier must be broken.

The swept domain decomposition rule[21] is a promising technology that breaks the latency barrier by communicating less often. The rule decomposes space and time into subdomains with swept boundaries, as opposed to static subdomains whose boundaries are straight in the time axis. The swept subdomains respect the domains of influence and dependency of the discretized PDE, making it possible to communicate once per many time steps. The swept domain decomposition enables simulations to be solved significantly faster than what is possible with static, straight decomposition.

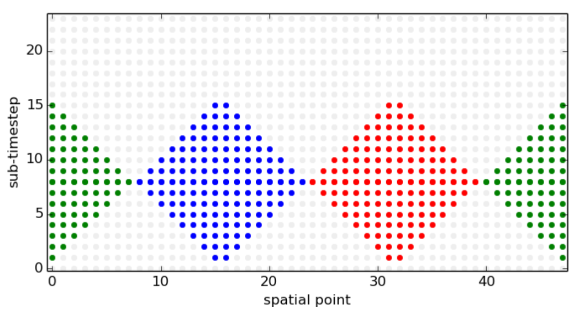
The swept decomposition rule works with finite difference, finite volume, and finite element methods. These methods discretize the spatial domain into a spatial graph; each point in the graph represents a grid point, a control volume, or an element; each edge represents that one grid is in the stencil of the other, two control volumes share an interface, or two elements share the support of a basis function. The swept rule can break the latency barrier in CFD solver components that require no global communication, including explicit time stepping and fixed-point iterations (e.g., red-black Gauss-Seidel). Such components can typically be decomposed into elementary sub-steps that require communication only between neighbors in the spatial graph. The swept rule decomposes the tensor product of discrete space (spatial graph) and discrete time (sub-steps), in a way that each subdomain, spanning many sub-steps, can be computed by a computing node without communicating with other nodes.



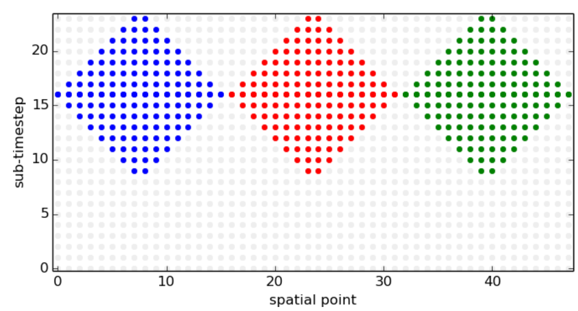
(a) Intersecting lines decompose discrete space-time



(b) Three computing nodes (different colors) process subdomains without communication



(c) Computing nodes process another set of subdomains after one communication



(d) Computing nodes process a third set of subdomains after the second communication

Figure 1: An illustration of the swept domain decomposition rule in 1D

Figure 1 illustrates how the swept rule works in one spatial dimension. Figure 1(a) illustrates how the discrete space-time domain can be decomposed using two arrays of points travelling in opposite spatial directions, resulting in the red and blue lines separating the decomposed domains. Figures 1(b)-(d) shows how computing nodes can advance many sub-steps with much fewer communications. In our experiment with the Kuramoto-Sivashinsky equation on Amazon EC2, about 20 sub-timesteps can be integrated during each latency time. In another experiment with the Euler equation on an Ethernet-based cluster, about 10 sub-timesteps can be integrated during each latency time.

Figure 2(a)-(f) show how the swept rule can work in two spatial dimensions, by using three arrays of lines travelling in three directions with obtuse angle to each other. The resulting three arrays of faces decomposes space-time into oblique cubes whose diagonal is aligned with the time dimension. This concept may be generalized to three dimensional space, with four arrays of faces travelling in four directions, forming four arrays of hyper-planes decomposing space-time into oblique hyper-cubes.

Research and development in swept decomposition rule can significantly benefit from a class of meshing

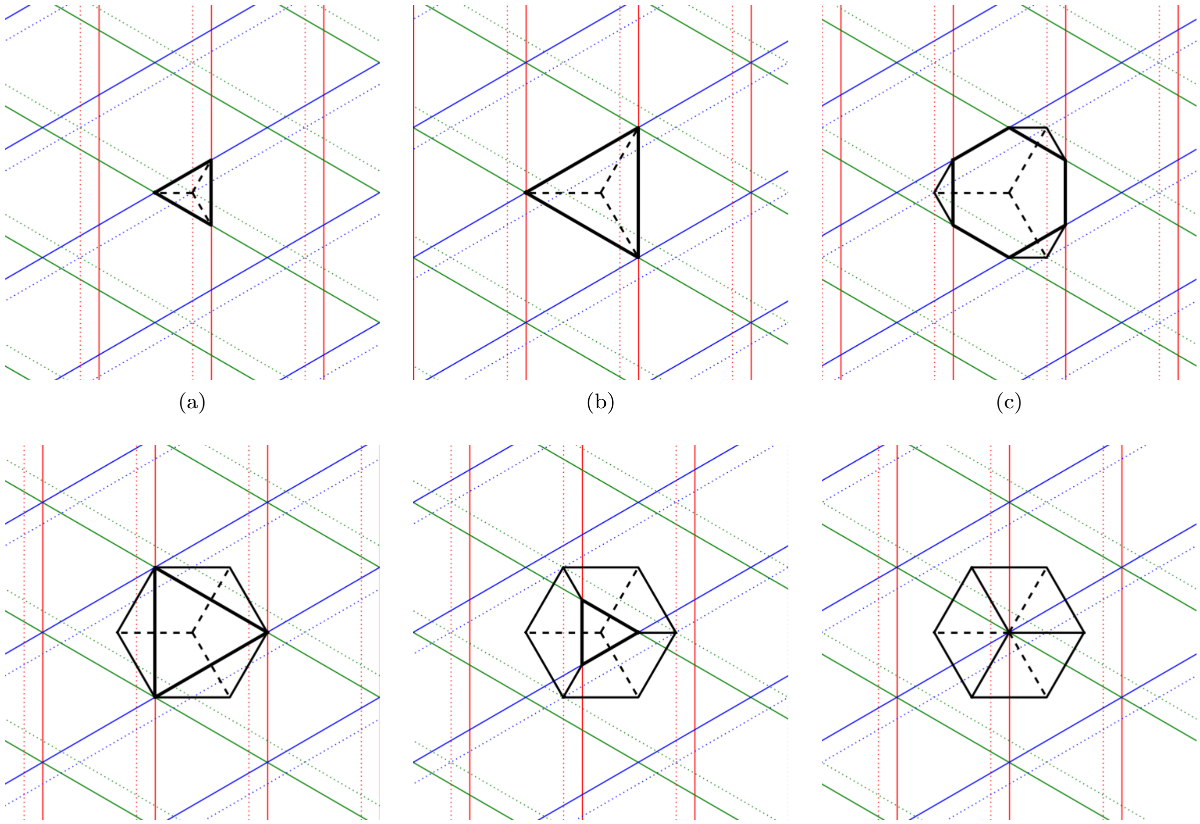


Figure 2: An illustration of the swept domain decomposition rule in 2D. Three arrays of lines, represented in red, green, and blue, travels in three directions with 120^o angle to each other. These sweeping lines slices the 2+1D space-time into oblique cubes. (a)-(f) illustrate how one such cube forms. Dotted, colored lines represent the position of the lines in the last frame. The partially formed cube in each frame is outlined in black; visible edges are in solid lines and edges behind the cube are in dashed lines.

algorithm for space-time discontinuous Galerkin method.[22–25] These meshing algorithms, e.g., the tent-pitcher algorithm,[26] partitions the continuous space-time into elements, such that the element boundaries respect the physical domain of influence and dependency. Most of these algorithms can be generalized to discrete space-time, leading to more scalable PDE solution algorithms that break the latency barrier.

IV. Towards faster high-fidelity design: Least squaares shadowing adjoint

To enable high-fidelity design in real time, not only the simulations but also the design process need to be significantly faster. including Large Eddy Simulations Essential to faster design is efficient optimization algorithm. A gradient-based algorithm runs far more efficiently than gradient-free algorithms for problems with many design variables. We can reduce the number of costly high fidelity simulations in optimization if we can compute the gradient of the objective function using the adjoint method.

The adjoint method is a powerful method in PDE-constrained optimization.[27,28] It traces sensitivities from the objective function backward to the design variables. This backward propagation of sensitivities is governed by the adjoint equation, which can be derived from the forward model using differentiation, chain rule and (in continuous adjoint) integration by parts. The solution to the adjoint equation can drive a gradient based optimization scheme to efficiently solve very high dimensional optimization problems in aerodynamic shape design and optimal control.[29,30]

Despite its potential, the adjoint method suffers from a fundamental problem when combined with chaotic nonlinear dynamics often observed in high fidelity CFD. Chaotic models are sensitive to initial conditions, a phenomenon popularly known as the "butterfly effect". A small perturbation to the equation can make an exponentially growing difference in its solution, until the differences saturated by nonlinearity in the model. Solutions to the adjoint equation suffer from the same exponential growth, only that the adjoint equation is linear, making the growth of its solution unbounded. When the adjoint equation is integrated over a long time period, such diverging solution can lead to a computed gradient that is orders of magnitude too large. Such wrong and useless gradients are computed from the adjoint solution even when the objective function consists of well-defined statistical or climate quantities, long time averages that are proven to be differentiable by Ruelles linear response theory.[31] As a result, it has only been applied to very limited cases of high fidelity simulations.

This divergent behavior of the adjoint is first analyzed by Lea et al.[32] in the context of climate modeling. The same divergence is discovered in chaotic, turbulent flow simulations by several investigators using a variety of flow solvers. In NASA Ames, the divergent adjoint was observed in 2D cylinder wakes, using an incompressible discontinuous Galerkin flow solver.[33] The divergent adjoint was observed in a separated airfoil, using NASA's flow solver FUN3D.[34]

This difficulty, manifesting as divergent adjoint, arises because of the sampling error. Unless a precisely defined initial condition is given, only infinite-long time averaged quantities in a chaotic system are completely predictable and therefore eligible as quantities of interest in engineering design. Any finite time average is less predictable in that it differs from its infinite-long time average counterpart by a small amount. This small amount, called the sampling error, behaves much like an instantaneous quantity in chaotic systems: it is unpredictable, apparently random; and is sensitive to small perturbations. The longer we time average a quantity, the smaller the sampling error we can expect it to have. Its rate of diminishing, however, is so slow that the sampling error is negligible in few practical simulations. Causing more trouble than its non-negligible size is its sensitivity to perturbations. A small change in the design can change the sampling error unpredictably. When performing sensitivity analysis on a finite time average, which is the sum of the infinite-long time average and the sampling error, the sensitivity of the sampling error often overwhelms the sensitivity of the infinite-long time average. When this happens, it is difficult to deduce the sensitivity of the infinite-long time average, which is the true quantity of interest, from the sensitivity of the finite time average, which is what can practically be computed from a simulation.

One way to overcome this problem is by filtering. The sampling error can be modeled as statistical noise that is independent for each design. Statistical filtering through a large number of designs would therefore remove the effect of the noise. Because it requires many simulations, however, filtering is expensive, to the point that it may not even be practical when the space of possible designs has many dimensions.

To overcome this problem, we must decouple the sensitivity of statistics from the realization of the flow field. A promising approach to this question is the shadowing approach.[35] The idea is to linearize the governing equation of a chaotic simulation, but not the initial condition. Liberated from the initial condition, the resulting linearized equation has many solutions. They describe how different the flow over a fixed pair of similar designs can be. Almost all these solutions diverge exponentially, but some of them do not. The ones that stay bounded are called *shadowing solutions* – they follow or "shadow" a given trajectory. Such shadowing simulations annot be obtained from the same governing law with different initial conditions, because the butterfly effect will cause them to have totally different snapshots after a while. Neither can shadowing be achieved by simulating different governing laws, such as different designs of a jet engine turbine blade, but starting from the same initial condition. The simulations would again diverge from each other. Shadowing can be achieved, however, by simultaneously perturbing both the governing law and the initial condition. The existence of such shadowing is guaranteed by the shadowing lemma of dynamical systems under strict conditions. The Least Squares Shadowing method finds such pairs of shadowing simulations, and performs sensitivity analysis with them. As the length of the simulation increases, not only does a time averaged quantity converge to the infinite time average, but also its computed derivative converges to the derivative of the infinite time average.[35]

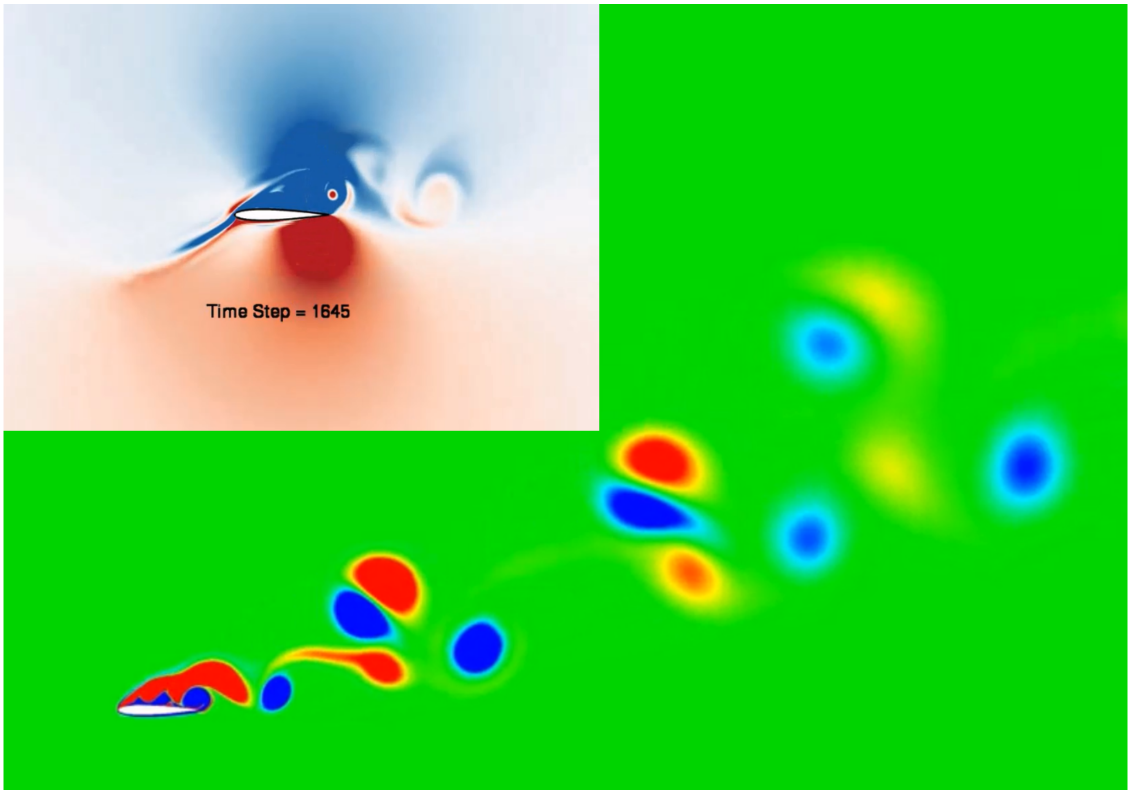


Figure 3: Chaotic vortex shedding over a stalled NACA 0012 airfoil (main) and the Least Squares Shadowing adjoint solution (inset)

Mathematically, the Least Squares Shadowing method replaces the initial value problem with the well-conditioned Least Squares Shadowing problem.[8, 10, 35] The solution to the Least Squares Shadowing problem satisfies the model equation, and give correct gradient that can drive efficient optimization algorithms. The Least Squares Shadowing method has been applied to a range of dynamical systems, the largest being an isotropic homogeneous turbulent flow simulation at Taylor microscale Reynolds number $Re_\lambda = 33$ by a 32^3 Fourier pseudo-spectral discretization.[36] It is also being implemented by NASA in its CFD solver FUN3D, and has produced preliminary solution on a chaotic 2D simulation of a stalled airfoil, shown in Figure 3.[37]

To enable high-fidelity design in real time, the Leasst Squares Shadowing method needs more research, both in how to efficiently compute the solution of the constrained least squares problem for large scale CFD applications, and in how to generalize to problems that are not strictly ergodic. If we can solve these challenges, the resulting adjoint sensitivity can quickly guide a designer towards better designs, even when a high-fidelity simulation reveals unfamiliar or nonintuitive flow physics. It would therefore allow engineers to better exploit larger variety of flow physics in their design, to achieve more aggressive design goals with innovative designs.

V. Conclusion

Computational simulation and design has much potential in aerospace engineering. To realize its potential, simulations need to be reliable and performed at higher fidelity. Such high fidelity simulation and its associated design must also be significantly accelerated in order for them to be broadly adopted. Many research directions are contributing to accelerating high-fidelity simulation and design. Because these contributions, together contributions from new ways of accelerating high-fidelity design, have multiplicative effect, we can hope to significantly accelerate high-fidelity design, perhaps to real-time for some aerospace applications.

Outlined in this paper are two relatively new approaches that can signifiantly accelerate high-fidelity design. The swept domain decomposition rule aims to break the latency barrier in massive parallel computing. It can lead to faster time integration of unsteady PDEs using larger number of computing nodes. The Least Squares Shadowing method aims to enable adjoint-based design for chaotic high-fidelity flow simulations. It can lead to significantly accelerated design for complex flows.

Combined with accelerations expected in other more established research fields, high-fidelity simulation and design can become widely adopted as a common practice in aerospace engineering. If such simulations are reliable and trusted, designing components for airplanes, launch vehicles and spaceships might become as easy as designing bicycles.

References

[1]Hicks, R. M. and Henne, P. A., "Wing design by numerical optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.

[2]Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD vision 2030 study: a path to revolutionary computational aerosciences," 2014.

[3]Hussaini, M., "ICASE: NASA Langley's CSE center," *Computational Science & Engineering, IEEE*, Vol. 2, No. 1, 1995, pp. 6–14.

[4]Johnson, F. T., Tinoco, E. N., and Yu, N. J., "Thirty years of development and application of CFD at Boeing Commercial Airplanes, Seattle," *Computers and Fluids*, Vol. 34, No. 10, 2005, pp. 1115 – 1151.

[5]Button, K., "Wanted: More focus on CFD," *Aerospace America*, January 2015, pp. 22.

[6]Brauckmann, G. J., Streett, C. L., Kleb, W. L., Alter, S. J., Murphy, K. J., and Glass, C. E., "Computational and Experimental Unsteady Pressures for Alternate SLS Booster Nose Shapes," *AIAA Science and Technology Forum and Exposition, Kissimmee, Florida*, 2015.

[7]Jones, W. T., Nielsen, E. J., Lee-Rausch, E. M., and Acree, C. W., "Multi-point Adjoint-Based Design of Tilt-Rotors in a Noninertial Reference Frame," *AIAA Science and Technology Forum and Exposition, National Harbor, Maryland*, 2014.

[8]Wang, Q., Gomez, S., Blonigan, P., Gregory, A., and Qian, E., "Towards Scalable Parallel-in-Time Turbulent Flow Simulations," *Physics of Fluids*, Vol. 25, No. 110818, 2013.

[9]Larsson, J. and Wang, Q., "The prospect of using large eddy and detached eddy simulations in engineering design, and the research required to get there," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 372, No. 2022, 2014, pp. 20130329.

[10]Wang, Q., Hu, R., and Blonigan, P., "Least Squares Shadowing sensitivity analysis of chaotic limit cycle oscillations," *Journal of Computational Physics*, Vol. 267, 2014, pp. 210–224.

[11]Ekaterinaris, J. A., "High-order accurate, low numerical diffusion methods for aerodynamics," *Progress in Aerospace Sciences*, Vol. 41, No. 3, 2005, pp. 192–300.

[12]Wang, Z., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H., et al., "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845.

[13]Spalart, P., Jou, W., Strelets, M., and Allmaras, S., "Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach," *Advances in DNS/LES*, Vol. 1, 1997, pp. 4–8.

[14]Piomelli, U. and Balaras, E., "Wall-layer models for large-eddy simulations," *Annual review of fluid mechanics*, Vol. 34, No. 1, 2002, pp. 349–374.

[15]Spalart, P. R., Deck, S., Shur, M., Squires, K., Strelets, M. K., and Travin, A., "A new version of detached-eddy simulation, resistant to ambiguous grid densities," *Theoretical and computational fluid dynamics*, Vol. 20, No. 3, 2006, pp. 181–195.

[16]Lyu, Z., Xu, Z., and Martins, J., "Benchmarking Optimization Algorithms for Wing Aerodynamic Design Optimization," *8th International Conference on Computational Fluid Dynamics (ICCFD8)*, 2014.

[17]Talnikar, C., Blonigan, P., Bodart, J., and Wang, Q., "Parallel Optimization for LES," *Proceedings of the Summer Program*, 2014, p. 315.

[18]Gropp, W., Kaushik, D., Keyes, D., and Smith, B., "Latency, bandwidth, and concurrent issue limitations in high-performance CFD," *Computational Fluid and Solid Mechanics*, 2000, pp. 839–841.

[19]Ding, C. and He, Y., "A ghost cell expansion method for reducing communications in solving PDE problems," *Supercomputing, ACM/IEEE 2001 Conference*, IEEE, 2001, pp. 55–55.

[20]Patterson, D. A., "Latency lags bandwith," *Communications of the ACM*, Vol. 47, No. 10, 2004, pp. 71–75.

[21]Alhubail, M. M. and Wang, Q., "The swept rule for breaking the latency barrier in time advancing PDEs," *arXiv preprint arXiv:1504.01380*, 2015.

[22]Lowrie, R. B., Roe, P. L., and Van Leer, B., "Space-time methods for hyperbolic conservation laws," *Barriers and Challenges in Computational Fluid Dynamics*, Springer, 1998, pp. 79–98.

[23]Abedi, R., Chung, S.-H., Erickson, J., Fan, Y., Garland, M., Guoy, D., Haber, R., Sullivan, J. M., Thite, S., and Zhou, Y., "Spacetime meshing with adaptive refinement and coarsening," *Proceedings of the twentieth annual symposium on Computational geometry*, ACM, 2004, pp. 300–309.

[24]Thite, S., "Adaptive spacetime meshing for discontinuous Galerkin methods," *Computational Geometry*, Vol. 42, No. 1, 2009, pp. 20–44.

[25]Erickson, J., Guoy, D., Sullivan, J. M., and Üngör, A., "Building spacetime meshes over arbitrary spatial domains," *Engineering with Computers*, Vol. 20, No. 4, 2005, pp. 342–353.

[26]Üngör, A. and Sheffer, A., "Tent-Pitcher: A meshing algorithm for space-time discontinuous Galerkin methods," *In proc. 9th intl. meshing roundtable*, Citeseer, 2000.

[27]Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, 1988, pp. 233–260.

[28]Reuther, J., Jameson, A., Alonso, J., Rimlinger, M., and Saunders, D., "Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers," *Journal of aircraft*, Vol. 36, No. 1, 1999, pp. 51–60.

[29]Bewley, T., Moin, P., and Temam, R., "DNS-based predictive control of turbulence: an optimal target for feedback algorithms," *J. Fluid Mech.*, Vol. 447, 2001, pp. 179–225.

[30]Wang, D., He, L., Li, Y., Wells, R., and Chen, T., "Adjoint Aerodynamic Design Optimization for Blades in Multi-Stage Turbomachines: Part II–Validation and Application," ASME, 2008.

[31]Ruelle, D., "A review of linear response theory for general differentiable dynamical systems," *Nonlinearity*, Vol. 22, No. 4, 2009, pp. 855.

[32]Lea, D., Allen, M., and Haine, T., "Sensitivity analysis of the climate of a chaotic system," *Tellus*, Vol. 52A, 2000, pp. 523–532.

[33]Barth, T., "Space-Time Error Estimation via Dual Problems for CFD," *SIAM Conference on Computational Science and Engineering*, SIAM, 2011.

[34]Blonigan, P. J., *New methods for sensitivity analysis of chaotic dynamical systems*, Master's thesis, Massachusetts Institute of Technology, 2013.

[35]Wang, Q., "Convergence of the Least Squares Shadowing Method for Computing Derivative of Ergodic Averages," *accepted for publication in SIAM J. Num. Anal.*, 2013, preprint available at arXiv:1304.3635.

[36]Gomez, S., *Parallel Multigrid for Large-Scale Least Squares Sensitivity*, Master's thesis, MIT, Cambridge, Massachusetts, USA, 2013.

[37]Nielsen, E. and Diskin, B., "Challenges in adjoint-based design for unsteady flows," *SIAM Conference on Computational Science and Engineering*, 2015.