Parallel Optimization for LES

By C. Talnikar†, P. Blonigan†, J. Bodart‡ AND Q. Wang†

We developed a parallel Bayesian optimization algorithm for large eddy simulations. These simulations challenge optimization methods because they take hours or days to compute, and their objective function contains noise as turbulent statistics that are averaged over a finite time. Surrogate based optimization methods, including Bayesian optimization, have shown promise for noisy and expensive objective functions. Here we adapt Bayesian optimization to minimize drag in a turbulent channel flow and to design the trailing edge of a turbine blade to reduce turbulent heat transfer and pressure loss. Our optimization simultaneously runs several simulations, each parallelized to thousands of cores, in order to utilize additional concurrency offered by today's supercomputers.

1. Introduction

In modeling turbulent flows, large eddy simulation (LES) can reliably capture flow separation and other phenomena that traditional models including Reynolds averaged Navier-Stokes (RANS) struggle with. Using LES for design and optimization is not only desirable, but also is made affordable by increasingly powerful computing clusters. Despite growing computing power, optimization with LES is algorithmically challenging. This is because of the unsteady, potentially chaotic dynamics of LES, which introduces a sampling error in the turbulent statistics (Oliver *et al.* 2014). Turbulent statistics are long-time averages of quantities of interest in a turbulent fluid flow.

$$\bar{J} = \mathrm{E}[J] = \lim_{T\to\infty} \frac{1}{T} \int_0^T J(u(t))dt. \tag{1.1}$$

The infinite time average is approximated by a sample average, which introduces a sampling error in the turbulent statistic. Because of chaotic dynamics, totally different sampling errors can come from LES of almost identical designs, starting from the same initial condition.

For performing optimization with LES we need to consider algorithms that are robust to noisy function evaluations. As computing gradients accurately in an LES is expensive we considered only derivative free optimiztion techniques (Rios & Sahinidis 2013). Surrogate-based methods work quite well for such problems as they filter out the noise in the evaluations. In recent years Bayesian optimization is emerging as a promising technique for optimizing noisy and expensive black box functions (Jones *et al.* 1998). It uses Gaussian processes for fitting a surrogate to the function evaluations. Successive evaluation points are decided by using a metric like expected improvement (EI).

A typical LES has two stages; the transient stage followed by the quasi-steady stage. Only during the quasi-steady stage are the statistics sampled. The transient stage, needed merely for reaching the quasi-steady stage, can be comparable to or longer than the quasi-steady state. To avoid wasting time on the transient stage of a new design, particularly

† Aerospace Computational Design Laboratory, Massachusetts Institute of Technology
‡ University of Toulouse ISAE, France

when it is similar to an old design that has already been simulated into quasi-steady stage, we may opt to continue simulating the old design. We call this technique snapping. In addition, the optimization runs in parallel by evaluating multiple designs at the same time, allowing it to scale to larger supercomputers. Bayesian optimization using Gaussian processes provides a good framework for formulating these ideas into an algorithm.

2. Bayesian optimization

Bayesian optimization fits function evaluations using a Gaussian process (GP). A GP is a collection of infinite random variables, defined by a mean function ($m(x)$) and a covariance function ($k(x, x')$). If $f(x)$ is the true function, then the GP is given by (Rasmussen & Williams 2006)

$$m(x) = \mathrm{E}[f(x)], \tag{2.1}$$

$$k(x, x') = \mathrm{E}[(f(x) - m(x))(f(x') - m(x'))], \tag{2.2}$$

One of the simplest choices for the covariance function is the squared exponential kernel ($k(x, x') = \exp[-(|x - x'|/c_l)^2]$). All realizations of the GP are smooth, infinitely differentiable, and vary over the length scale c_l.

A GP can explicitly model noise in the function evaluations. The noise in an LES is the sampling error, the variance of which can be estimated (Oliver *et al.* 2014). During optimization, some designs have noisier objective functions than other designs. GPs provide a way to incorporate this information in the fitting process. Consider a set of sample points x_* having the function evaluations y_*, with variance σ_*^2. The evaluations y, σ^2 at x can be found from the GP using the following formula

$$y(x) = k_*^T (K + \sigma_*^2 I)^{-1} y_*, \tag{2.3}$$

$$\sigma^2(x) = k(x_*, x_*) - k_*^T (K + \sigma_*^2 I)^{-1} k_*, \tag{2.4}$$

where $K = k(x, x), k_* = k(x, x_*)$. The hyperparameters including the characteristic length c_l can be decided using a Bayesian model selection.

Once the GP is constructed, we must decide what designs to simulate next. This decision must fulfil two competing goals: exploration and exploitation. Exploration means the objective should be evaluated in regions where the uncertainty(noise) is high to improve the quality of the surrogate. Exploitation means the objective should be evaluated at the minimum of the surrogate to get the precise value of the optimum. A metric that provides a good balance between the two is expected improvement (EI). It is defined by the following formula

$$\mathrm{EI}(x) = \mathrm{E}[\max(f_{min} - Y(x), 0)], \tag{2.5}$$

where f_{min} is the current best value of the objective and Y is the random variable corresponding to point x. EI has a compact analytical form by computing the expectation as an integral (Snoek *et al.* 2012)

$$\mathrm{EI}(x) = (f_{min} - y(x))\Phi\left(\frac{f_{min} - y(x)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{f_{min} - y(x)}{\sigma(x)}\right), \tag{2.6}$$

where ϕ is the standard normal density and Φ is the distribution function. The point in the design space, which maximizes EI, is chosen as the next evaluation point.

EI works well for uniformly noisy objective functions, but as we know in an LES the objectives have heterogeneous noise, hence EI needs to be adapted for such cases. A recent proposal is to multiply EI by a penalization function to discount points that have already been evaluated. It is known as augmented expected improvement (AEI) and the multiplier is given by (Picheny *et al.* 2013)

$$M(x) = 1 - \frac{\tau}{\sqrt{\sigma^2(x) + \tau^2}}, \tag{2.7}$$

where τ is a tunable parameter and σ is the standard deviation at a point x evaluated from the GP. A more rigorous formulation is the expected quantile improvement (EQI). Because the evaluations are noisy, choosing f_{min} in EI to be the current best evaluation can be misleading. EQI instead defines improvement on the basis of minimum β-quantile $q_n(x) = y(x) + \Phi^{-1}(\beta)\sigma(x)$. EQI is defined as

$$\mathrm{EQI}(x) = \mathrm{E}[\max(\min(q_n(x)) - q_{n+1}(x), 0)], \tag{2.8}$$

EQI can also be computed analytically and the expression turns out to be quite similar to EI, except that the mean and the variance are conditional.

Today's computers are massively parallel, with more computing cores than the strong scaling limit of many simulations. To utilize additional concurency, we want our optimization to simulate multiple designs in parallel. Bayesian optimization using EI, however, is inherently serial. At each step EI is maximized, a single point is evaluated and the process is repeated. The EI criterion needs to be modified to support maximization over multiple points, so that multiple points can be evaluated at the same time in parallel. A natural extension is the following (Ginsbourger *et al.* 2008)

$$\mathrm{EI}(x_1, x_2, ..., x_n) = \mathrm{E}[\max(f_{min} - \min(Y(x_1), Y(x_2)..., Y(x_n)), 0)], \tag{2.9}$$

A problem with the above expression is that it cannot be computed analytically for $n > 2$ and requires expensive Monte Carlo evaluations. The practical approach is to use an approximation. To avoid computing joint distributions needed for multi-point EI, a sequential multi-point EI can be performed. The basic idea is to successively do 1-point EIs by conditioning them on the point computed in the previous step. This still requires a value for the objective at the new points; a safe strategy is to simply use the mean of the GP at that point. This is known as the Kriging believer strategy and the main problem is that the optimizer may get trapped in non optimal regions if the initial GP fit is bad. An alternative is to use the constant liar strategy in which a constant value (L) is used as the objective evaluation at all the points computed in the multi-point EI. Choices for L are $\min(y), \max(y)$. Higher L leads to more explorative optimizers.

Apart from exploration and exploitation, in LES it is also important to consider whether to continue a previous evaluation. If instead of simulating a new design, a simulation at an old design is extended we say that the new design has snapped on to the old design. This will reduce the uncertainty at the design point and can improve the quality of the GP fit. It is also much cheaper to extend an evaluation than to start a new one because of the large transient times in a turbulent fluid flow. There are many possible criteria that can be used to decide when to snap to a previous evaluation or start a new one. One idea is to check if the two evaluation points are close by. If the Euclidean distance between the two points is below a certain threshold then the new point is snapped onto the previous one. A problem with this method is that it cannot be non-dimensionalized easily and a certain amount of tuning might be required for each case. Also in higher dimensions the probability of snapping drastically reduces as the design space is much larger. An alternative is to check if the points are close by using the relative difference between EI at the two points. There is still a certain amount of

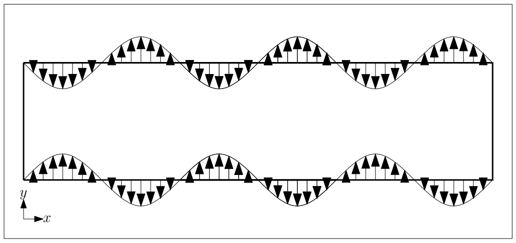

Figure 1. Diagram of the traveling wave boundary conditions used for drag reduction. The z direction points out of the page.

tuning to be done, but this idea follows the principle of evaluating points that lead to an improvement in the optimal and surrogate fit.

Before starting the optimization it is important to have a good set of evaluations from which the initial surrogate is created. For the design of experiment Latin hypercube sampling was used to find the design points for the initial evaluations.

3. Turbulent channel drag reduction

To demonstrate our parallel Bayesian optimization algorithm, we consider flow control for drag reduction of a turbulent channel. Specifically, we consider the traveling waves studied by Min *et al.* (2006), Moarref & Jovanović (2010), and Lieu *et al.* (2010). By enforcing sinusoidal inflow/outflow at the walls as shown in Figure 1, the Reynolds shear stresses can be modified in the near wall region, resulting in changes to the drag of the channel. It has been shown that for certain amplitudes, if these waves are made to propagate upstream, the channel drag will be smaller than that for laminar flow-through the same channel with the same mass flow rate (Min *et al.* (2006)).

We have used our optimization algorithm to find the wave speed c for which the most drag reduction is achieved for a given mass flow rate. To this end, we used a time-averaged fractional change in drag as an objective function

$$J = \frac{\Delta D}{D_{baseline}} = \frac{D - D_{baseline}}{D_{baseline}},$$

where D is the time-averaged drag of the channel and $D_{baseline}$ is the time-averaged drag of the same channel with no flow control. This makes the channel optimization a one-dimensional, single-objective optimization problem.

Simulations were conducted with the compressible LES solver CharLES^x. The mass flow was kept constant with an additional momentum forcing term. This term was chosen so that the bulk velocity is 0.2 units. This corresponds to a Reynolds number of Re = 13800 with channel height as the length scale, much larger than that used by Min *et al.* (2006).

The channel geometry used is 6 × 2 × 3 units, with periodic boundary conditions in the x and z directions. At the walls, the x and z velocities are fixed, while the y velocity had the following form

$$v(x) = \pm a \cos\left(\frac{\pi}{3}(x - ct)\right),$$

where a, the wave amplitude is positive on the lower wall and negative on the upper wall. For our optimization, a was fixed at 0.04 units, or 20% of the channel bulk velocity.

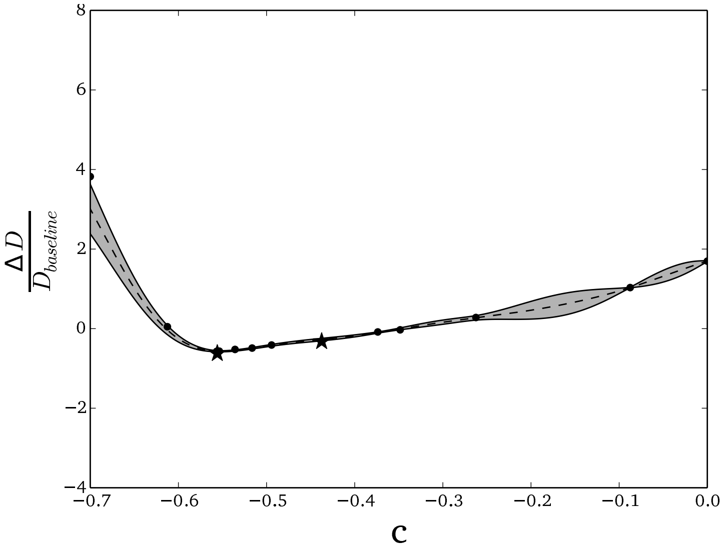

FIGURE 2. GP fit for the change in drag from the baseline (no-control) case $\Delta D/D_{baseline}$ versus wave speed c. Negative values of c correspond to upstream traveling waves. The region bounded by one standard deviation σ of the GP is shaded grey. (- - - -) GP mean,(●) evaluations, (★) snapped evaluations.

A 128 × 128 × 128 structured mesh of quads is used, where the cell sizes are uniform in the x and z directions. In the y direction, the cell nearest to the wall has an edge length of 0.0015 units.

The simulations were run for roughly 200 time units for each function evaluation. This includes 100 time units, or 20 flow-through times, of transient data that is ignored when computing the objective function $\Delta D/D_{baseline}$. The objective function is computed from the momentum source term used to drive the channel flow, which is available as an output from CharLES^x.

Two key features of our algorithm, parallelism and snapping were demonstrated in finding the optimal wave speed for drag reduction in a turbulent channel. For each iteration of the optimization algorithm, two function evaluations were done in parallel. Snapping was enabled but only used twice.

The optimization results are shown in Figure 2. The optimal wave speed was found to be $c = -0.5703$ (the wave speed c is non-dimensionalized as wave velocity divided by the speed of sound), resulting in around 60% drag reduction from the baseline case with no traveling wave. Note the large number of function evaluations in the vicinity of the minimum, including the two points where snapping was used at $c = -0.4375$ and $c = -0.5590$. This shows that the decisions made using EI result in an effective use of computational resources.

The trend in fractional drag change is consistent with that found by Min *et al.* (2006) for low-magnitude values of wave speed c. As wave speed increases in magnitude upstream, the drag decreases through the changes in the Reynolds stresses near the wall. Eventually, the Reynolds shear stress $\bar{u'v'}$ becomes negative, and the drag falls to sub-laminar values near the optimum (drag of a laminar channel with no control is 55% smaller than that of a turbulent channel with no control Min *et al.* (2006)). The increase in drag to the left of the optimum was not observed in literature, and arises from com-

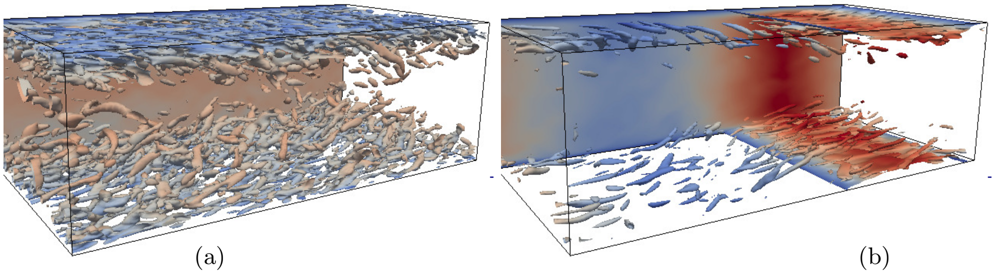

FIGURE 3. Q criterion isocontours colored by streamwise-velocity for uncontrolled (a) and optimally controlled (b) channel flow. The contours correspond to $Q = 0.04$ in both plots.

pressibility effects. At $c = -0.6$ and less, acoustic waves dominate the flow field, which is not surprising considering the nearly transonic speeds attained by the traveling waves.

Finally, we consider the vortical flow structures shown in Figure 3 for the baseline and optimized cases. Comparing these plots, we see very few structures in the optimized change relative to the baseline case. In modifying the Reynolds stresses near the walls, the traveling waves have destroyed a large number of flow structures.

Also, it is important to note the sheets on the channel walls. These sheets correspond to the inflow and outflow boundary conditions: the sheet begins just downstream of the maximum inflow velocity and ends near the maximum outflow velocity. One can think of the boundary conditions as a vortex sheet, where one period of the sinusoid, inflow followed by outflow just downstream, represents one vortex. The Q-criterion reveals the presence of these vortices at the walls with the sheets.

4. Turbine blade shape design

The second test case for the Bayesian optimization formulation is the trailing edge design of a turbine blade for minimizing the heat transfer and pressure loss. The blade is a turbine nozzle guide vane designed by researchers at Von Karman Institute (VKI) (Arts & de Rouvroit 1992). The baseline case for this optimization is the blade design given in the aforementioned paper. The chord length of the blade is 67.647mm and the blades are in a linear cascade with the pitch being 0.85 times the chord length. The inlet in the simulation setup of the problem is 100mm upstream of the leading edge of the blade. The inlet flow is at an angle of 55 degrees to the chord of the blade. A linear cascade is simulated by having periodic boundary conditions at the top and bottom. The spanwise extent of the setup is 10mm. The blade surface on the pressure and suction side is assumed to be isothermal, the inlet isentropic Mach number is 0.9 and the isentropic Reynolds number downstream is 10^6. The turbulent intensity at the inlet is 4% and the length scale is 3mm. The compressible flow solver used is CharLES which is an explicit finite volume code. Perturbations to the mean flow at the inlet are injected using a synthetic turbulence generator.

The flow accelerates as it goes around the suction side and reaches a peak of Mach 1. At the specified turbulent intensity the boundary layer transitions from a laminar to turbulent at the suction side about 60mm along the curved surface starting from the leading edge. The flow separates at the trailing edge. The shape of the trailing edge can greatly influence the seperation locations on the pressure and suction side as well as flow in the recirculation region. This affects the heat transfer on the surface of the blade and the pressure loss in the flow. The objective of the optimization is set to be a linear combination of the non-dimensional versions of the two. The heat transfer is characterized by the Nusselt number

$$Nu = \frac{\bar{h}L}{k}, \tag{4.1}$$

where k is the thermal conductivity at $T = 300K$, 0.0454 W/mK. L is the trailing edge radius in the baseline case, 0.71mm. $\bar{h}$ is the average heat transfer coefficient integrated over time and the part of the blade from 28mm downstream of the leading edge up to the trailing edge. The formula for $\bar{h}$ is

$$\bar{h} = \frac{1}{S(t_j - t_i)\Delta T} \int_{t_i}^{t_j} \int_S k \frac{\partial T}{\partial n} dS dt, \tag{4.2}$$

where t_i is the transient time and t_j is the stop time of the simulation. S is the surface area. $\Delta T = 120K$ is the temperature difference between the surface of the blade and the stagnation temperature of the flow.

The pressure loss is characterized by the pressure loss coefficient. It is computed as the pressure loss divided by the inlet total pressure $\bar{p}_l = \bar{p}_{t,l}/p_{t,in}$. The pressure loss is computed 16mm downstream of the trailing edge by cutting a plane normal to the inlet flow. The formula for $\bar{p}_{t,l}$ is

$$\bar{p}_{t,l} = \frac{1}{S(t_j - t_i)} \int_{t_i}^{t_j} \int_S (p_{t,in} - p_{t,p}) dS dt \tag{4.3}$$

where $p_{t,in}$ is the inlet total pressure and $p_{t,p}$ is the total pressure at the plane given by $p_p[1+(\gamma-1)M_p^2/2]^{\frac{\gamma}{\gamma-1}}$, M_p is the Mach number and p_p is the static pressure at the plane. The time integration for both these quantities is started after an initial transient time. For this simulation it was chosen to be 0.3 times the time it takes for a single flow-through based on the amount of time it takes for the objective to stabilize. Time averaging was then performed over 0.7 flow-through time. The weights in the linear combination of the two objective functions were chosen such that both were of similar magnitude for the baseline case and less than one.

The mesh for the simulation is a hybrid structured and unstructured mesh as shown in Figure 4(a). The smallest cell size away from the walls is 0.5mm, which is about a factor of 6 smaller than the most significant eddies. The first cell size at the wall is set to 0.02mm, which results in a maximum y_+ of 10 over the surface of the blade to reduce the cost of the simulation (Collado Morata *et al.* 2012). This means that the mesh for the boundary layer is under-resolved, and a wall model is required for reliably capturing the heat transfer characteristics at the blade surface. A wall model for CharLES which works for blades having a high flow incidence angle, is currently under development. In the mean time, the simulation was run without a wall model to validate the performance of our optimizer.

To parameterize the trailing edge in two dimensions we used B-splines. They are piecewise polynomial functions of a specific order that can be used to represent curves. To limit the number of parameters needed to characterize the curve, the order of the B-spline was chosen to be 3 with 2 control points as shown in Figure 4(b). By moving these points we can change the shape of the B-spline. Two control points translate to 4 degress of freedom. The range of the parameters (locations of the 2 control points) was chosen such that the trailing edge is not too sharp and there is no loop present in the curve. For

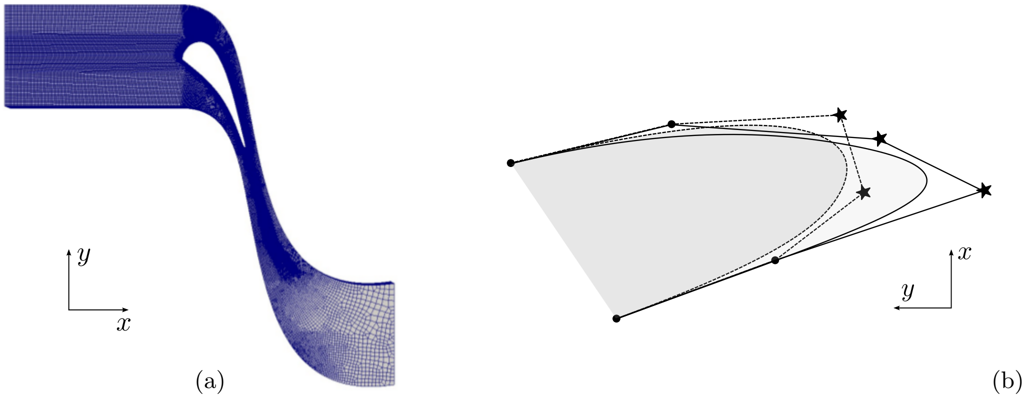

FIGURE 4. (a) Computational mesh around the baseline geometry. (b) Baseline (- - - -) and optimal(———) trailing edges. (●) indicates fixed points and (★) control points used in the design.

the actual meshing, a set of points defining the entire blade (fixed part + trailing edge curve) is passed to the ANSYS ICEM meshing tool that generates the mesh. The mesh topology is assumed to be constant as the changes to the shape of the blade are local and restricted to the trailing edge.

The choices of objective function and design parameters make the blade optimization a four-dimensional, single-objective optimization problem. In this optimization, 10 points were evaluated in the design of experiment and 25 points were evaluated using the parallel EQI criterion. The number of points evaluated in parallel were 4. The value of the objective in the baseline case is 0.6877 (Nusselt number: 6604, Pressure loss coefficient: 0.02241) with a standard deviation of 0.0140. The value of the objective for the optimal case (Figure 4(b)) is 0.5587 (Nusselt number: 5486, pressure loss coefficient: 0.01775). We get a 17% reduction in heat transfer and a 21% reduction in pressure loss. During the optimization run, the criterion twice snapped onto the final design, i.e., it chose to continue simulating this design instead of starting to simulate a similar design proposed by EQI.

In the Figure 5, we compare the flow fields near the trailing edge of the baseline and optimal case. From Figure 5(e), we see that the optimal design is skewed toward the pressure side. This design ensures that separation on the pressure side is considerably downstream of the separation on the suction side, unlike the baseline case.

The optimizer moves the separation point downstream on the pressure side because the boundary layer is laminar. Because of this, the mixing layer between the recirculation region and the fluid on the pressure side is much thinner than that on the suction side. This allows the hot fluid outside the pressure side boundary layer to enter the recirculation region with relative ease, as is evident in the temperature contour shown in Figure 5(f). Once this fluid enters the recirculation region, it flows back towards the trailing edge, resulting in the higher heat fluxes at the trailing edge. By moving the pressure side separation point downstream, the mixing layer is moved downstream, and as a result the portion of the mixing layer adjacent to the recirculation region is more stable. This results in less mixing and less hot fluid in the recirculation region, which in turn reduces the heat transfer to the trailing edge.

At the same time, the separation point on the suction side is moved upstream, and a

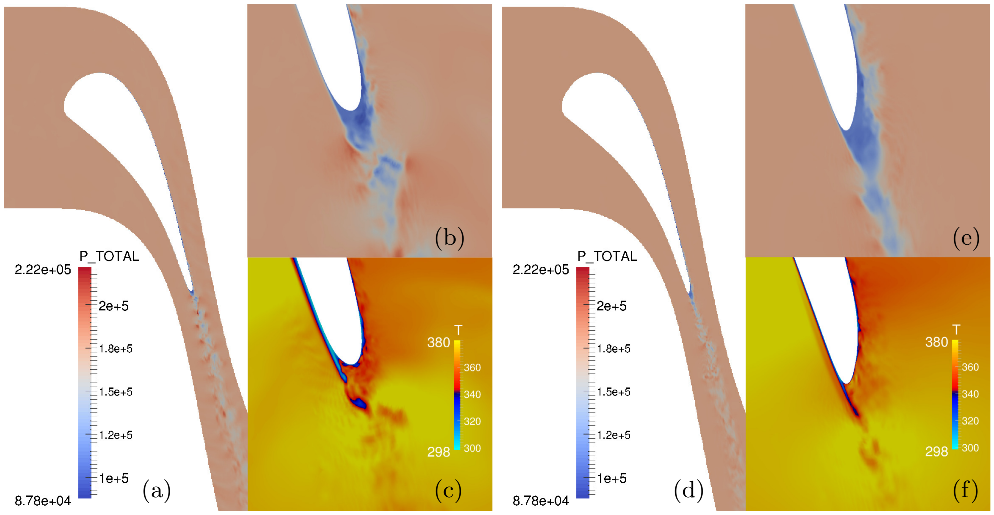

FIGURE 5. (a) Baseline geometry with total pressure contours in Pascals (b) Zoom in the trailing edge region. (c) Map of the temperature in Kelvins. (d-e-f) Same plots for the optimal geometry.

larger portion of the trailing edge is exposed to the recirculation region. However, because the boundary layer on the suction side is turbulent, the fluid in this region is well mixed. Because the fluid near the wall is much cooler than the fluid outside the boundary layer, the fluid in the turbulent boundary layer and the portion of the recirculation region adjacent to the suction side of the blade is relatively cool.

Additionally, the spacing between the separation regions on the pressure and suction side affects the wake structure considerably. In Figure 5(a), we can see coherent structures in the total pressure contours indicating vortices shed from the trailing edge for the baseline case. However, these distinct structures are absent from the optimized design. The absence of these vortex structures implies less large-scale mixing, which is consistent with the considerable reduction in pressure loss that is achieved by the optimized design.

5. Conclusion

In summary, we have demonstrated a novel parallel Bayesian optimization framework for flow control for a turbulent channel flow and for the design of the trailing edge of a turbine inlet guide vane. The framework uses a parallel expected quantile improvement (EQI) criterion to determine whether to explore new designs or to exploit existing designs to reduce the local error in design space by continuing existing LES. Furthermore, it can explore designs in parallel, allowing the use of as many computational resources as desired.

In both test cases our framework found a vastly improved design from the baseline case. To find these optimized designs, the framework employed novel features such as snapping and a parallel function evaluations.

Given the promising results of our parallel Bayesian optimization framework, we plan to apply it to the turbine blade case presented above with a working wall model. Additionally, we plan to apply our framework to other applications, to demonstrate its utility to wider range of engineers and scientists.

Acknowledgments

We thank Jasper Snoek and Ryan Adams from Harvard University for sharing their expert knowledge on Bayesian optimization. Mihailo Jovanovic and Armin Zare from University of Minnesota have contributed to optimization of the travelling wave in turbulent channel. The turbine trailing edge optimization problem is motivated by Gregory Laskowski and Michael Hayek from GE Aviation and Sriram Shankaran from GE Global Research. Sanjeeb Bose and Sanjiva Lele from Stanford University and Gaofeng Wang from Zhejiang University helped formulate the trailing edge optimization problem. The first and last authors acknowledge funding from GE Aviation and GE Global Research. The LES are performed on Cetus/Mira supercomputers at Argonne National Lab, using the CharLES and CharLES^x solver. We particularly thank the CTR Summer Program for making possible this massively collaborative work.

REFERENCES

Arts, T. & de Rouvroit, M. L. 1992 Aero-thermal performance of a two-dimensional highly loaded transonic turbine nozzle guide vane: A test case for inviscid and viscous flow computations. *J. Turbomach.* **114**, 147–154.

Collado Morata, E., Gourdain, N., Duchaine, F. & Gicquel, L. 2012 Effects of free-stream turbulence on high pressure turbine blade heat transfer predicted by structured and unstructured les. *Int. J. Heat. Mass. Tran.* **55**, 5754–5768.

Ginsbourger, D., Le Riche, R. & Carraro, L. 2008 *A multi-points criterion for deterministic parallel global optimization based on Gaussian processes.* In Hal Archives-Ouvertes, Report no. hal-00260579.

Jones, D. R., Schonlau, M. & Welch, W. J. 1998 Efficient global optimization of expensive black-box functions. *J. Global Optim.* **13**, 455–492.

Lieu, B. K., Moarref, R. & Jovanović, M. R. 2010 Controlling the onset of turbulence by streamwise travelling waves. part 2. direct numerical simulation. *J. Fluid Mech.* **663**, 100–119.

Min, T., Kang, S., Speyer, J. & Kim, J. 2006 Sustained sub-laminar drag in a fully developed channel flow. *J. Fluid Mech.* **558**, 309–318.

Moarref, R. & Jovanović, M. R. 2010 Controlling the onset of turbulence by streamwise travelling waves. part 1. receptivity analysis. *J. Fluid Mech.* **663**, 70–99.

Oliver, T., Malaya, N., Ulerich, R. & Moser, R. 2014 Estimating uncertainties in statistics computed from direct numerical simulation. *Phys. Fluids* **26**.

Picheny, V., Ginsbourger, D., Richet, Y. & Caplin, G. 2013 Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics* **55**, 2–13.

Rasmussen, C. & Williams, C. 2006 *Gaussian Processes for Machine Learning*. MIT Press.

Rios, L. M. & Sahinidis, N. V. 2013 Derivative-free optimization: A review of algorithms and comparison of software implementations. *J. Global Optim.* **56**, 1247–1293.

Snoek, J., Larochelle, H. & Adams, R. P. 2012 Practical bayesian optimization of machine learning algorithms. In *25th Neural Processing Systems Proceedings*, pp. 2951–2959.