Towards an adjoint based 4D-Var state estimation for turbulent flow

Pieter Bauweraerts and Johan Meyers
Celestijnenlaan 300A, B3001 Leuven, Belgium
E-mail: pieter.bauweraerts@kuleuven.be, johan.meyers@kuleuven.be

Abstract. The turbulent nature of the atmospheric boundary layer leads to wind-turbine power fluctuations and fatigue loading. The possibility to perform faster than real-time turbulent flow field simulations opens up applications in the forecasting and potentially actively controlling of turbulent structures. An important aspect of the forecasting, is the initialization of the simulation based on measurements, which will be the focus of this work. To this end, we use a 4D-Var approach, which is already being applied in numerical weather prediction applications. In this article, a fully developed pressure-driven atmospheric boundary layer is used as a case study. As a underlying model for the state estimation, we use a large eddy simulation (LES) code run on a relatively coarse grid, to decrease the model evaluation time. We take virtual measurements from a reference LES simulation, which is also used to benchmark the state estimation. For the 4D-Var optimization problem, we use a L-BFGS approach combined with an adjoint LES simulation for the gradient calculations.

1. Introduction

Turbulence in the Atmospheric Boundary Layer (ABL) plays an important role in many natural processes and engineering applications. In recent years, LIDAR has emerged as a new technology that can scan turbulent wind fields over areas of several kilometers. This opens up many new opportunities, including the use of measured wind fields for real-time turbulent flow forecasting in the atmospheric boundary layer, with prediction horizons ranging from a couple of minutes up to one hour. This can have applications in power forecasting for short-term trading or grid services, and can potentially be used to control the turbulent wind fields in a wind farm to maximize power, track a predefined power signal or reduce fatigue loading.

An important aspect is the evaluation speed of the flow-field model. This needs to be multiple times faster than real time, such that the state-estimation algorithm, which typically needs several flow-field evaluations, is faster than real-time. For wind-farm applications, wake models are typically used [1, 2], these do not include atmospheric turbulence, but rather predict mean-flow statistics. However, recently in ref. [3] the feasibility of coarse grid large eddy simulations (LES) as a real-time flow field model has been investigated.

Even with LIDAR measurements, the amount of flow-field measurements remains in practice significantly smaller than the amount of flow-field states. To this end, we use the 4D-Var data assimilation strategy, which has previously been applied in numerical weather-prediction applications [4]. Other research has focused on the retrieval of flow field structures, see e.g. ref. [5] and [6], but with a offline calculated viscosity model, instead of a full subgrid scale (SGS) model.
2. Methodology

2.1. Flow field model

The simulations in the current manuscript are performed using an inhouse pseudo-spectral LES code SP-Wind, which has been developed at KU Leuven over the last decade [7]. For our work, we approximate the ABL, using an incompressible pressure driven BL, which is governed by the following Navier–Stokes equations

\[
\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} - \frac{1}{\rho} \nabla (p + p_{\infty}) - \nabla \cdot \tau_{\text{SGS}},
\]

\[
\nabla \cdot \mathbf{u} = 0,
\]

where \( \mathbf{u} = (u_1, u_2, u_3) \) represents the filtered velocity field, \( \nabla p_{\infty} = [f_{\infty}, 0, 0] \) represents the background pressure gradient which we presume in the \( x \)-direction, and \( p \) the remaining pressure after subtracting the background pressure \( p_{\infty} \). We neglect contributions of the viscous stresses on the resolved scales, and model the subgrid scale stresses \( \tau_{\text{SGS}} \) using a Smagorinsky model [8]

\[
\tau_{\text{SGS}} = 2l^2 (2\mathbf{S} : \mathbf{S})^{1/2} \mathbf{S},
\]

with \( \mathbf{S} = (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)/2 \) the rate-of-strain tensor. For the Smagorinsky length scale \( l \), we employ Mason’s [9] damping function, such that \( l^n = (C_s \Delta)^{-n} + (\kappa(z + z_0))^n \), where \( \Delta = (\Delta_x \Delta_y \Delta_z)^{1/3} \) is the local grid spacing, and \( C_s \) the Smagorinsky constant, and \( \kappa = 0.4 \) the Von Kármán constant. We take \( C_s = 0.14, n = 1 \), see ref. [10] for a discussion.

For the boundary conditions, we use periodicity in the horizontal directions, symmetry at the top of the domain, and impermeability in combination with a wall stress model, which is applied to the first grid point at the bottom (see ref. [11]). This gives the following equation for the wall stress at the first grid point

\[
\tau_{w,1} = -\left( \frac{\kappa}{\log(z_1/z_0)} \right)^2 \left( \tilde{u}_1^2 + \tilde{u}_2^2 \right) \tilde{u}_1,
\]

\[
\tau_{w,2} = -\left( \frac{\kappa}{\log(z_1/z_0)} \right)^2 \left( \tilde{u}_1^2 + \tilde{u}_2^2 \right) \tilde{u}_2,
\]

where the parallel velocity components \( \tilde{u}_1 \) and \( \tilde{u}_2 \) are obtained from filtering the horizontal velocities \( u_1, u_2 \) with a 2D Gaussian filter, using filter widths \( 4\Delta_x \) and \( 4\Delta_y \) in \( x \) and \( y \)-direction. Further \( z_1 \) is the vertical coordinate of the first grid point, while \( z_0 \) is the surface roughness.

SP-wind uses a pseudo-spectral discretization for the horizontal directions. The nonlinear terms are evaluated in real space and dealiased using the 3/2 rule [12]. Fourier transforms are performed using the FFTW library [13]. For the vertical direction, a 4th-order energy conservative scheme is used [14]. The parallelization is performed using a 2D pencil decomposition of the Fourier transforms [15]. For the time integration, a 4th-order explicit Runge–Kutta scheme is employed. The time step is chosen by applying the Courant-Friedricht-Lewis (CFL) condition \( \max(\mathbf{u} \Delta t / \Delta) \leq C_{\text{max}} \), where \( C_{\text{max}} = 0.4 \) [16].

2.2. Optimization problem formulation

We aim to penalize the squared difference between measured and the simulated velocities, respectively \( \mathbf{u}_{\text{meas}} \) and \( \mathbf{u} \). This leads to the following cost function \( J \)

\[
J(\mathbf{u}) = \int_{-T}^{0} \int_{\Gamma} \frac{1}{2} (\mathbf{u}_{\text{meas}} - \mathbf{u})^2 \, dx \, dt,
\]
where \( \Gamma \) represents a subset of the domain \( \Omega \) where measurements are available, and \( T \) the time window of measurements used for the optimization. Note that in this preliminary study, we use a strongly simplified form of a 4D-var approach, where for simplicity, we do not consider measurement and model uncertainties, or contributions in the cost function due the result of state estimations of previous time horizons.

The optimization problem can then be summarized as

\[
\begin{aligned}
\text{minimize} & \quad J(u), \\
\text{subject to} & \quad \frac{\partial u}{\partial t} + u \cdot \nabla u + \frac{1}{\rho} \nabla (p + p_\infty) + \nabla \cdot \tau_\text{SGS} = 0 \quad \text{in } \Omega \times T, \\
& \quad \nabla \cdot u = 0 \quad \text{in } \Omega \times T, \\
& \quad u(x, 0) = u_0(x) \quad \text{in } \Omega. 
\end{aligned}
\]

Note that directly solving the optimization problem in the current format leads to an optimization problem which has order \( O(N_x N_y N_z N_t) \) optimization variables, which within current standards of LES simulations quickly adds up to billions of optimization variables. To circumvent this, the optimization problem is rewritten in a reduced formulation

\[
\begin{aligned}
\text{minimize} & \quad \tilde{J}(q(\Psi; t)), \\
\text{subject to} & \quad \tilde{\nabla}_\text{xy} \cdot \hat{u} = \hat{k}_x \hat{u}_1 + \hat{k}_y \hat{u}_2, \\
& \quad \tilde{\nabla}_\text{xy} \times \hat{u} = -i\hat{k}_y \hat{u}_1 + i\hat{k}_x \hat{u}_2,
\end{aligned}
\]

where \( \tilde{J} \) is the reduced cost function, \( \Psi \) are the independent unconstrained control variables, and \( u = q(\Psi; t) \) is the implicit solution operator of the NS-equations, which transforms the control variables to the velocity field at different \( t \). In the next sections this methodology is further elaborated.

### 2.3. Parametrization of the initial field

To avoid having constraints or penalization terms in the cost function to enforce a divergence free initial flow field \( u_0 \), an alternative representation is derived in this section. First of all, the continuity equation is rewritten as

\[
\nabla_{xy} \cdot u + \frac{\partial u_3}{\partial z} = 0,
\]

where for the left hand side the divergence is taken in the xy-plane, to this end a 2D nabla operator is defined as \( \nabla_{xy} = \left\{ \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, 0 \right\} \). We additionally introduce the z-component of the vorticity as

\[
\nabla_{xy} \times u = \omega_3.
\]

Using Helmholtz’s theorem, we can reconstruct a vector field based on its divergence and curl. This implies that we can reconstruct our horizontal components \( u \) and \( v \) of our velocity field \( u \) based on equations 9 and 10. Note that our code uses a spectral discretization in xy direction (see section 2.1) such that in spectral space the horizontal derivatives are equal to a multiplication with their respective wave numbers and the imaginary unit \( i \), this gives

\[
\begin{aligned}
\nabla_{xy} \cdot \hat{u} &= ik_x \hat{u}_1 + ik_y \hat{u}_2, \\
\nabla_{xy} \times \hat{u} &= -ik_y \hat{u}_1 + ik_x \hat{u}_2,
\end{aligned}
\]

here the hat denotes the variables in Fourier space \( \hat{u} = \mathcal{F}(u) \), and \( k_x \) and \( k_y \) are the streamwise and spanwise wave numbers. Writing the previous equations in matrix form gives

\[
\begin{bmatrix}
0 & 0 & I \\
-ik_y & ik_x & 0 \\
-ik_x & -ik_y & D_{zz}
\end{bmatrix}
\begin{bmatrix}
\hat{u}_1 \\
\hat{u}_2 \\
\hat{u}_3
\end{bmatrix} =
\begin{bmatrix}
\hat{u}_3 \\
\hat{\omega}_3 \\
0
\end{bmatrix}
\]

\( \mathcal{F} \)
where \( D_z \) is introduced as the discrete derivative in z direction. This matrix can be easily inverted analytically

\[
\begin{bmatrix}
\hat{u}_1 \\
\hat{u}_2 \\
\hat{u}_3
\end{bmatrix} = \frac{1}{k_x^2 + k_y^2} \begin{bmatrix}
-k_y D_z & -i k_y I & i k_y I \\
-k_x D_z & i k_x I & 0 \\
(1 + k_x^2)I & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\hat{u}_3 \\
\hat{\omega}_3
\end{bmatrix}
= \frac{1}{k_x^2 + k_y^2} \begin{bmatrix}
-k_y D_z & -i k_y I \\
-k_x D_z & i k_x I \\
(1 + k_x^2)I & 0
\end{bmatrix}
\begin{bmatrix}
\hat{u}_3 \\
\hat{\omega}_3
\end{bmatrix}
= T \hat{\Psi},
\]

such that the vertical flow field \( \hat{u}_3 \) and the vertical vorticity component \( \hat{\omega}_3 \) give a full representation of the velocity field \( u \). Thus we introduce the free parameters \( \hat{\Psi} = \{ \hat{u}_3, \hat{\omega}_3 \} \) of the initial flow field, and the transformation matrix \( T \) to transform the parameter set to the full velocity field. Note that this inversion does not work for both \( k_x \) and \( k_y \) being zero, which represent the mean flow at a certain xy-plane. The continuity equation in this case reduces to \( \partial U_3(z)/\partial z = 0 \), where \( U_3 \) is the mean vertical velocity, which implies a constant vertical velocity \( U_3(z) = Cte \). Since, we have no outflow boundary conditions at the top and bottom of our domain, this constant has to be zero, and is thus omitted for the optimization. For other vertical boundary conditions, this constant can simply be added to the optimization problem. The stream- and spanwise mean velocities on the other hand, respectively defined as \( U_1(z) \) and \( U_2(z) \), are used directly as control variables.

### 2.4. Optimization methodology

For the optimization we use a limited-memory BFGS solver as is further described in ref. [17]. The number of optimization variables is \( O(N_x N_y N_z) \), which makes it computationally prohibitively expensive to use a finite difference gradient calculation. However, by using an adjoint approach, the gradient to all control variables can be calculated by solving an additional set of linear partial differential equations, the adjoint equations, at the computational cost of approximately one forward LES run (see e.g. [18] for an introduction). The derivation of the adjoint equations for our problem is very similar to the one described in ref. [19], and is for sake of brevity not repeated here. The final result is given by

\[
-\frac{\partial \xi}{\partial t} - u \cdot \nabla \xi - (\nabla \xi)^T \cdot u - \nabla \cdot \tau_{SGS} - \nabla \pi + f_{SE} = 0,
\]

\[
\nabla \cdot \xi = 0,
\]

where \( \xi \) is the adjoint velocity field, \( \pi \) the adjoint pressure term, \( \tau_{SGS} \) the adjoint subgrid scale stresses and \( f_{SE} \) the adjoint forcing term appearing due to the cost function \( \mathcal{J} \). The forcing term can easily be shown to be equal to

\[
f_{SE} = \begin{cases} 
    u_{\text{meas}} - u & \text{for all } x \in \Gamma \\
    0 & \text{for all } x \notin \Gamma
\end{cases}
\]

The adjoint equations are integrated backward in time from \( t = 0 \) to \( t = -T \), with the initial conditions \( \xi(x; 0) = 0 \). The gradient to the control variables is given by

\[
\tilde{\mathcal{J}}_{\Psi}(\delta \Psi) = \int_{\Omega} \mathcal{T}^* \xi(x; -T) \cdot \delta \Psi \, dx,
\]

where \( \mathcal{T}^* \) is the conjugate transpose of \( \mathcal{T} \).
2.5. Case study
For our case study, we use a pressure driven boundary layer. The streamwise length, width and height are respectively taken to be $6.4 \times 3.2 \times 1 \text{ km}$. Since this is a feasibility study, we keep the amount of grid cells limited, to keep the computational cost limited. The grid specifications are summarized in table 1. For the optimization we use a time horizon $T$ of 200 s.

For simplicity, we take measurements over the whole domain, such that $\Gamma = \Omega$. A more realistic implementation of virtual sensors like e.g. LIDARS will be the subject of future work. We take the measurements from a fully developed turbulent BL, such that the turbulent statistics remain approximately constant in time. For the first iteration, we initialize the controls for the optimization with zero ($\Psi_0 = 0$).

3. Results
3.1. Adjoint gradient validation
The gradient obtained by the adjoint approach is verified by finite differences. For the finite differences a step length of $1 \times 10^{-6}$ is empirically found as a good trade-off between round off errors and non-linearities. Figure 1 shows a vertical slice of the gradients for the different control variables, and a selection of wave numbers (see eq. 19, with $\delta \Psi$ a unit perturbation to one of the control parameters). It is noted that there is in general a very good agreement and that the gradients are spread over a wide variety of scales. Only at the small scales, where gradients tend to be very small anyway, discrepancies start to show.

3.2. Case study
In this section the main results of the case study are presented. For the optimization we initialize all our controls with zero. First, we define the error as follows

$$
\epsilon = \left| \frac{J_i - J_\infty}{J_0} \right|
$$

(20)

where $J_i$ is the cost function after $i$ iterations, and $J_\infty$ is the cost function after full convergence note that this is theoretically zero in our case. On figure 2 the error is shown as a function of the amount of function and gradient evaluations. A quick initial decrease is noted, due to the mean flow values settling relatively quickly, which have a big impact on the cost function. Afterwards, a monotonous decline is observed.

When looking at the flow fields, figure 3 shows the control field after 900 function and gradient evaluations. It is noted that the big scales have converged relatively well, this in contrast with the small scales which are underrepresented in the solution. The exact cause needs further investigation, but possible causes are firstly, the slower convergence due to the relative low sensitivity in comparison with larger scales. Secondly, numerical errors in the adjoint equations,
Figure 1: Vertical slices of the adjoint (triangle) and finite difference (circle) gradients $\nabla \tilde{J}_\Psi(\delta \Psi)$ for a selection of control parameters $\Psi$. Figures (a) and (b) respectively show the gradients to the mean flow $U_1$ and $U_2$. Figures (c) and (d) for the gradients to the real part of the vertical component of vorticity $Re(\hat{\omega})$ respectively for the wave numbers $(k_{x,\text{min}}, 0)$ and $(10k_{x,\text{min}}, 5k_{y,\text{min}})$. Figures (e) and (f) then finally show the same but for $Re(\hat{u}_3)$ instead of $Re(\hat{c})$. 

(a) 

(b) 

(c) 

(d) 

(e) 

(f)
Figure 2: The convergence of the optimization is shown. The vertical axis shows the normalized error $\epsilon$ and the horizontal axis shows the amount of function and gradient evaluations.

Figure 3: The left hand side (a) represents the estimated control field based on the measurements and after 450 optimization iterations, the middle (b) shows the exact reference field, the right hand side (c) finally shows the difference between the two. The top figures each time shows an xz-slice and the lower figures show a xy-slice at $z=400m$ which introduce errors in the gradients, which can significantly slow down or even prevent convergence.

4. Conclusion
In this work, a 4D-Var state-estimation algorithm is tested using virtual measurements, combined with a coarse grid LES flow field model. The adjoint gradient calculation is verified, and the methodology is demonstrated on a pressure driven atmospheric boundary-layer as a simple test case. We stress that this study is conceptual, and in future work we want to improve the flow field measurements such that they resemble more closely to reality. This will need the addition of regularization terms to the cost function to keep a well posed optimization problem, due to locations where flow information is absent or limited. Moreover a comparison with a more accurate reference, in the form of measurements of a finer grid simulation would be an interesting addition.
Acknowledgements
The authors acknowledge support from the Agency for Innovation by Science and Technology in Flanders (IWT). The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government department EWI.

References
[1] Katic I, Højstrup J and Jensen N O 1986 A simple model for cluster efficiency European wind energy association conference and exhibition pp 407–410
[2] Niayifar A and Porté-Agel F 2016 Energies 9 741
[3] Bauweraerts P and Meyers J Manuscript in review
[4] Talagrand O and Courtier P 1987 Quarterly Journal of the Royal Meteorological Society 113 1311–1328
[5] Chai T, Lin C L and Newsom R K 2004 Journal of the atmospheric sciences 61 1500–1520
[6] Lin C L, Chai T and Sun J 2001 Journal of the atmospheric sciences 58 1767–1783
[7] Calaf M, Meneveau C and Meyers J 2010 Physics of fluids 22 015110
[8] Smagorinsky J 1963 Monthly weather review 91 99–164
[9] Mason P J and Thomson D J 1992 Journal of Fluid Mechanics 242 51–78
[10] Meyers J 2011 Journal of Scientific Computing 49 65–77
[11] Moeng C H 1984 Journal of the Atmospheric Sciences 41 2052–2062
[12] Canuto C, Hussaini M Y, Quarteroni A, Thomas Jr A et al. 2012 Spectral methods in fluid dynamics (Springer Science & Business Media)
[13] Frigo M and Johnson S G 2005 Proceedings of the IEEE 93 216–231
[14] Verstappen R and Veldman A 2003 Journal of Computational Physics 187 343–368
[15] Li N and Laizet S 2010 2decomp & fft-a highly scalable 2d decomposition library and fft interface Cray User Group 2010 conference pp 1–13
[16] Courant R, Friedrichs K and Lewy H 1967 IBM journal of Research and Development 11 215–234
[17] Zhu C, Byrd R H, Lu P and Nocedal J 1997 ACM Transactions on Mathematical Software (TOMS) 23 550–560
[18] Borzí A and Schulz V 2011 Computational optimization of systems governed by partial differential equations vol 8 (SIAM)
[19] Goit J P and Meyers J 2015 Journal of Fluid Mechanics 768 5–50