Computational study of the effect of rotation in a model of the turbulent flow due to a cylinder rolling along ground

[Turbulent flow past a rolling cylinder]

Johan Hoffman

School of Computer Science and Communication, KTH, SE-10044 Stockholm, Sweden

Abstract

We consider the problem of computational simulation of the flow past a wheel of a vehicle. As a model we use a rotating cylinder in contact with a flat surface moving with the same velocity as the uniform free stream. In particular we are interested in the importance of including the effect of rotation in the model to accurately compute the drag force. We compare two different models; (i) a cylinder rolling around ground, and (ii) a stationary cylinder in a free stream, corresponding to a simple wind tunnel testing. We compute the drag for (i) and (ii) at a Reynolds number of 10 000, and the results are compared. We find that the flow field past the rotating cylinder is fundamentally different from the flow field past the non-rotating cylinder, with almost twice as large drag. The computations are carried out using a General Galerkin method, where the mesh is being adaptively refined to approximate the drag coefficient $c_D$.

Key words: adaptive finite element method, turbulent flow, large eddy simulation, cylinder rolling along ground, drag reduction

PACS: 47.27.Eq, 47.11.+j, 02.70.Dh

Preprint submitted to Elsevier Science 14th February 2008
1 Introduction

We consider the computational modelling of the turbulent flow past the wheels of a car, or other vehicle. Of particular interest is minimization of the drag of the vehicle, which is directly coupled to the fuel consumption, and since typically a significant portion of the total drag is attributed to the wheels, reducing the drag of the wheels could give a major overall drag reduction. Thus it would be very valuable to be able to accurately predict the drag of different wheel designs, and in particular to understand the influence on drag when using simplifications of the full model. In this paper we focus on the effect of the rotation of the wheels, and what the effect on drag is for not including realistic rotation in the model.

Experimentally, drag of a vehicle is determined from wind tunnel testing, and a crucial problem here is to recreate the conditions of a moving vehicle. A first approximation is to place a stationary vehicle on the floor of the wind tunnel, ignoring the effects of (i) the rotation of the wheels, and (ii) the creation of an artificial boundary layer at the ground surface. This is a simple and still very common way to model the flow around a car [Hucho and Sovran 1993]. Next, various methods to reduce the effect (ii) of the boundary layer is used, such as boundary layer suction or tangential blowing, but still not representing the effect (i) of the rotation of the wheels. The more sophisticated wind tunnels use a moving belt in combination with suction, where the wheels are driven by the moving belt. One seems to be able to recreate realistic road conditions to a certain extent using this approach, although this technique requires considerable model preparation, with separate suspensions for the wheels and the vehicle, and the method is not well suited for routine testing but is primarily restricted to research [Hucho and Sovran 1993].
An alternative approach to determine the aerodynamic properties of a vehicle design is *Computational Fluid Dynamics CFD*, which does not suffer from the limitations of wind tunnel testing, and which has many other benefits; such as being able to generate information of a certain design before even a testable model is built, flexibility to easily modify the model, better possibilities to model realistic situations since disturbencies from the wind tunnel is avoided (walls, testing devices, etc.), and downscaling of the model (such as an airplane) is not necessary. In particular, it appears to be easier to take (i)-(ii) into consideration using CFD. In addition, today the creative design process primarily takes place on the computer, and thus information (such as drag) of a certain design before building a model would be very valuable.

In this paper our aim is to use CFD to investigate the importance of including (i)-(ii) in a model to recreate realistic conditions. As computational model we consider the flow due to a cylinder rolling along a flat surface, modeling the flow past the wheels of a car or other vehicle. More specifically, we consider the flow past totally exposed wheels such as racing car wheels or airplane wheels during take-off or landing. Of course, there are many simplifications in this model, such as not including the wheel axes and using a simplified geometry, but here we focus on the effect of (i)-(ii).

For CFD to be competitive as a development tool, the computations need to be accurate and cheap, that is we need a computational method that is *reliable* and *efficient*. CFD concerns the computational solution of the Navier–Stokes equations (NSE), which are considered to model both laminar and turbulent flow. A *Direct Numerical Solution (DNS)* of NSE is very expensive, since the number of mesh points in space-time needed to represent all the physical scales in the flow may be estimated to be $\sim \text{Re}^3$, where $\text{Re} = UL/\nu$ is the Reynolds number, $L$ and $U$
are characteristic length and velocity scales, and \( \nu \) is the viscosity. The Reynolds number in many applications of interest is larger than \( 10^6 \), and thus more than \( 10^{18} \) mesh points would be needed, which is far beyond the capacity of any computer of today.

The traditional approach to get around the impossibility of DNS for high Reynolds numbers is to seek new equations satisfied by certain mean values of the NSE solution. This is typically accomplished by averaging the NSE, which introduce so-called Reynolds stresses, from averaging the non linear term in NSE. The averaging may be global corresponding to Reynolds Averaged Navier-Stokes equations (RANS), or more local in space corresponding to Large Eddy Simulation (LES). The Reynolds stresses depend on the non averaged NSE solution, and thus need to be modelled in terms of the averaged solution in a turbulence model, referred to as the problem of closure. The closure problem is the major open problem of CFD, with the existing models being highly dependent of the particular problem under consideration, and what numerical method that is being used. Quantitative error control is in general not available, and thus accuracy cannot be guaranteed.

In [Hoffman and Johnson 2006b, Hoffman 2005a] the General Galerkin G2 method is introduced as an adaptive computational method for turbulent flow, where a stabilized Galerkin finite element method is used to compute a chosen mean value output to a specified tolerance, using a minimal number of degrees of freedom. The solution is computed from NSE directly, without introducing any Reynolds stresses by filtering the equations. In this way the closure problem is circumvented, in the sense that no Reynolds stresses need to be modelled. Instead the stabilization in G2 acts as a simple turbulence model.

G2 is based on a posteriori error estimation, where the output sensitivity informa-
tion is obtained by computational approximation of an associated dual problem, linearized at an approximate G2 solution, and with data coupling to the output of interest. In particular, both the error from discretization and the error from the stabilization are controlled a posteriori. For an overview of adaptive finite element methods including many references, we refer to the survey articles [Eriksson et al. 1995, Becker and Rannacher 2001, Giles and Süli 2002, Hoffman and Johnson 2004]. The generalization of these methods to turbulent flow is first presented in [Hoffman 2004, Hoffman and Johnson 2006b, Hoffman 2005a], with applications to flow around a surface mounted cube and a square cylinder, and the flow past a sphere and a circular cylinder is investigated in [Hoffman 2005b, Hoffman 2006a]. For further material on G2 we refer to the new book [Hoffman and Johnson 2006a].

Many computational studies are available for rotating cylinders in a free stream, but the number of computational studies investigating the influence of the contact with the ground, which is extremely important in many applications, seems to be less common. In [Bhattacharyya et al. 2004] a computational model in 2d for laminar flow is investigated, and in this paper we extend this type of model to turbulent flow in 3d.

We use G2 to compute the drag coefficient $c_D$ for a cylinder rolling along a flat surface at $Re = 10^4$, and we also compute $c_D$ for the case of a stationary cylinder on a flat surface in a free stream, corresponding to testing in a wind tunnel without a moving belt. We find that the two flows are fundamentally different, with the drag of the rotating cylinder being almost twice as large.

In this paper we first recall the Navier-Stokes equations and the G2 method in sections 2-6, and we then describe the computational model in Section 7 before concluding with computational results in sections 8-9.
The incompressible Navier-Stokes equations expressing conservation of momentum and incompressibility of a Newtonian fluid with constant kinematic viscosity \( \nu > 0 \) enclosed in a volume \( \Omega \) in \( \mathbb{R}^3 \) (where we assume that \( \Omega \) is a polygonal domain) with no slip (homogeneous Dirichlet) boundary conditions for the velocity, take the form: Find \( \mathbf{u} = (u, p) \) such that

\[
\begin{align*}
\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \Delta \mathbf{u} + \nabla p &= f \quad \text{in } \Omega \times I, \\
\nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega \times I, \\
\mathbf{u} &= 0 \quad \text{on } \partial \Omega \times I, \\
\mathbf{u}(\cdot, 0) &= \mathbf{u}_0 \quad \text{in } \Omega,
\end{align*}
\]

where \( \mathbf{u}(x, t) = (u_i(x, t)) \) is the velocity vector and \( p(x, t) \) the pressure of the fluid at \( (x, t) \), and \( f, \mathbf{u}_0, I = (0, T) \), is a given driving force, initial data and time interval, respectively. The quantity \( \nu \Delta \mathbf{u} - \nabla p \) represents the total fluid force, and may alternatively be expressed as

\[
\nu \Delta \mathbf{u} - \nabla p = \text{div } \mathbf{\sigma}(\mathbf{u}, p),
\]

where \( \mathbf{\sigma}(\mathbf{u}, p) = (\sigma_{ij}(\mathbf{u}, p)) \) is the stress tensor, with components \( \sigma_{ij}(\mathbf{u}, p) = 2\nu \epsilon_{ij}(\mathbf{u}) - p \delta_{ij} \), composed of the stress deviatoric \( 2\nu \epsilon_{ij}(\mathbf{u}) \) with zero trace and an isotropic pressure: here \( \epsilon_{ij}(\mathbf{u}) = (u_i_{,j} + u_j_{,i})/2 \) is the strain rate tensor, with \( u_i_{,j} = \partial u_i / \partial x_j \), and \( \delta_{ij} \) is the usual Kronecker delta, the indices \( i \) and \( j \) ranging from 1 to 3. Assuming that (1) is normalized so that the reference velocity and typical length scale are both equal to one, the Reynolds number \( \text{Re} \) is equal to \( \nu^{-1} \).
The cG(1)cG(1) method is a General Galerkin G2 method [Hoffman and Johnson 2006a], using the continuous Galerkin method cG(1) in space and time. With cG(1) in time the trial functions are continuous piecewise linear and the test functions piecewise constant. cG(1) in space corresponds to both test functions and trial functions being continuous piecewise linear. Let \(0 = t_0 < t_1 < \ldots < t_N = T\) be a sequence of discrete time steps with associated time intervals \(I_n = (t_{n-1}, t_n]\) of length \(k_n = t_n - t_{n-1}\) and space-time slabs \(S_n = \Omega \times I_n\, \), and let \(W^n \subset H^1(\Omega)\) be a finite element space consisting of continuous piecewise linear functions on a mesh \(\mathcal{T}_n = \{K\}\) of mesh size \(h_n(x)\) with \(W^n_0\) the functions \(v \in W^n\) satisfying the Dirichlet boundary condition \(v|_{\partial \Omega} = \mathbf{w}\).

We seek \(\hat{\mathbf{U}} = (\mathbf{U}, P)\), continuous piecewise linear in space and time, and the cG(1)cG(1) method for the Navier-Stokes equations (1), with homogeneous Dirichlet boundary conditions reads: For \(n = 1, \ldots, N\), find \((\mathbf{U}^n, P^n) \equiv (\mathbf{U}(t_n), P(t_n))\) with \(\mathbf{U}^n \in V^n_0 \equiv [W^n_0]^3\) and \(P^n \in W^n\), such that

\[
((\mathbf{U}^n - \mathbf{U}^{n-1})k_n^{-1} + \mathbf{U}^n \cdot \nabla \mathbf{U}^n, \mathbf{v}) + (2\nu \epsilon(\mathbf{U}^n), \epsilon(\mathbf{v})) - (P^n, \nabla \cdot \mathbf{v})
+ (\nabla \cdot \mathbf{u}^n, q) + SD_\delta(\mathbf{U}^n, P^n; \mathbf{v}, q) = (f, \mathbf{v}) \quad \forall \mathbf{v} = (\mathbf{v}, q) \in V^n_0 \times W^n,
\]

where \(\bar{\mathbf{U}}^n = \frac{1}{2}(\mathbf{U}^n + \mathbf{U}^{n-1})\), with the stabilizing term

\[
SD_\delta(\mathbf{U}^n, P^n; \mathbf{v}, q) \equiv (\delta_1(\mathbf{U}^n \cdot \nabla \mathbf{U}^n + \nabla P^n - f), \mathbf{U}^n \cdot \nabla \mathbf{v} + \nabla q) + (\delta_2 \nabla \cdot \mathbf{U}^n, \nabla \cdot \mathbf{v}),
\]

and \(\delta_1 = \frac{1}{2}(k_n^{-2} + |\mathbf{U}|^2 h_n^{-2})^{-1/2}\) in the convection-dominated case \(\nu < |\bar{\mathbf{U}}^n|h_n\) and \(\delta_1 = \kappa_1 h_n^2\) otherwise, \(\delta_2 = \kappa_2 h_n\) if \(\nu < |\bar{\mathbf{U}}^n|h_n\) and \(\delta_2 = \kappa_2 h_n^2\) otherwise, with \(\kappa_1\)
and $\kappa_2$ positive constants of unit size (in this paper we have $\kappa_1 = \kappa_2 = 1$), and

$$(v, w) = \sum_{K \in T_n} \int_K \mathbf{v} \cdot \mathbf{w} \, dx,$$

$$(\epsilon(v), \epsilon(w)) = \sum_{i,j=1}^{3} (\epsilon_{ij}(v), \epsilon_{ij}(w)).$$

We note that the viscous term $(2\nu \epsilon(U), \epsilon(v))$ may alternatively occur in the form $(\nu \nabla U, \nabla v) = \sum_{i=1}^{3} (\nu \nabla(U)_i, \nabla v_i)$. In the case of Dirichlet boundary conditions the corresponding variational formulations are equivalent, but not so in the case of Neumann boundary conditions. If we have Neumann boundary conditions, we use the standard technique to apply these boundary conditions weakly.

### 4 Solution of the discrete system

To compute a cG(1)cG(1) solution $\hat{U}$, we have to solve a nonlinear system of algebraic equations at each time-step. We solve the system for $(U^n, P^n)$ on slab $S_n$ using a fix-point iteration with the convection velocity given by the previous iteration. Assuming the nodal values $(U^{n,j}, P^{n,j})$ in iteration $j$ have been computed, we compute new nodal values $(U^{n,j+1}, P^{n,j+1})$ by solving a linearized version of cG(1)cG(1) of the form:

$$
A U^{n,j+1} + k_n B P^{n,j+1} = k_n F^n,
$$

$$
B^T U^{n,j+1} + C P^{n,j+1} = G^n,
$$

where $A = M_n + k_n N_n - k_n \nu \Delta_n$, where $M_n$ is a mass matrix, the matrix $N_n$ represents a discrete analog of the convection term with velocity $U^{n,j}$, $\Delta_n$ represents a discrete Laplacian, $B$ is a discrete gradient, $B^T$ a discrete divergence, $C = -\delta_1 \Delta_n$, and finally $F^n$ and $G^n$ represent terms given by data including $U^{n-1}$.

We here first solve for $P^{n,j+1}$ in terms of $U^{n,j}$ in the second equation using a multi-
grid method, and then solve for $U^{n,j+1}$ in the first equation using GMRES. The resulting fixed point iteration converges under a CFL-condition (that is if $\frac{|U^n|k_n}{h_n} < 1$ is small enough), and if also $k_n/\delta_1$ is small enough. Since typically $\delta_1 \approx h_n/|U^n|$, convergence is thus achieved under a CFL-condition. In the applications in this paper of non-stationary high Reynolds number turbulent flow, convergence is usually obtained in 1-2 iterations, assuming the CFL-condition is satisfied.

5 Computation of drag

Using partial integration, the mean value in time of the drag $N(\sigma(\mathbf{u}))$ of a body may be expressed as [Giles et al. 1997, Hoffman and Johnson 2006a]:

$$N(\sigma(\mathbf{u})) = \frac{1}{|I|} \int_I (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f}, \Phi) - (p, \nabla \cdot \Phi) + (2\nu \epsilon(\mathbf{u}), \epsilon(\Phi)) + (\nabla \cdot \mathbf{u}, \Theta) dt,$$

(5)

where $\Phi$ is a function defined in the fluid volume $\Omega$, being equal to a unit vector $\phi$ in the direction of the flow on $\Gamma_0$, the surface of the body in contact with the fluid, and zero on the remaining part of the boundary $\Gamma_1 = \partial \Omega \setminus \Gamma_0$ of the fluid volume. The representation (5) is independent of $\Theta$, and the particular extension of $\Phi$ away from the boundary.

Here we think of $\mathbf{u} = (\mathbf{u}, p)$ as being a solution to (1) in the fluid volume $\Omega$ surrounding the body (using suitable boundary conditions as specified below), with sufficient regularity for the target output $N(\sigma(\mathbf{u}))$ in (5) to be well defined.

We are led to compute an approximation of the drag from a cG(1)cG(1) solution $\hat{\mathbf{U}} = (\mathbf{U}, P)$, using the formula
\[ N^h(\sigma(\bar{U})) = \frac{1}{|I|} \int_I (U_t + U \cdot \nabla U - f, \Phi) - (P, \nabla \cdot \Phi) \]
\[ + (2\nu e(U), \epsilon(\Phi)) + (\nabla \cdot U, \Theta) + SD_\delta(U, P; \Phi, \Theta) \, dt, \tag{6} \]

where now \( \Phi \) and \( \Theta \) are finite element functions (with as before \( \Phi = \phi \) on \( \Gamma_0 \) and \( \Phi = 0 \) on \( \Gamma_1 \)), and where \( U_t = (U^n - U^{n-1})/k_n \) on \( I_n \). We note the presence of the stabilizing term \( SD_\delta \) in (6), compared to (5), which is added in order to obtain the independence of \( N^h(\sigma(\bar{U})) \) from the choice of \((\Phi, \Theta), \) given by (3).

We define the drag coefficient \( c_D \) as a global average of a normalized drag force on the cylinder from the flow. We seek to approximate \( c_D \) by \( \bar{c}_D, \) a normalized drag force averaged over a finite time interval \( I \) at fully developed flow, defined by

\[ \bar{c}_D \equiv \frac{1}{2U_\infty^2 A} \times N(\sigma(\bar{u})), \tag{7} \]

where \( U_\infty = 1 \) is the free stream velocity, \( A = D \times D = D^2 \) is the cylinder area facing the mean flow, and \( N(\sigma(\bar{u})) \) is defined by (5). In computations we approximate \( \bar{c}_D \) by \( \bar{c}_D^h, \) defined by

\[ \bar{c}_D^h = \frac{1}{2U_\infty^2 A} \times N^h(\sigma(\bar{U})), \tag{8} \]

with \( N^h(\sigma(\bar{U})) \) being defined by (6). Thus we may use a scaled version of the a posteriori error estimate (10) below to estimate the error \( |\bar{c}_D - \bar{c}_D^h| \).

6 Adaptive cG(1)cG(1)

We now present an adaptive algorithm with a posteriori error estimation based on duality, where we introduce the following dual problem: Find \((\varphi, \theta) \) with \( \varphi = \phi \)
on $\Gamma_0$ and $\varphi = 0$ on $\Gamma_1$, such that

$$-\varphi_t - (u \cdot \nabla)\varphi + \nabla U \cdot \varphi - \nu \Delta \varphi + \nabla \theta = 0, \quad \text{in } \Omega \times I,$$

$$\nabla \cdot \varphi = 0, \quad \text{in } \Omega \times I,$$

$$\varphi(\cdot, T) = 0, \quad \text{in } \Omega,$$

(9)

where $(\nabla U \cdot \varphi)_j = \partial U / \partial x_j \cdot \varphi$.

Replacing the exact dual solution $(\varphi, \theta)$ by a computed approximation $(\varphi^h, \theta^h)$, we are led to the following a posteriori output error estimate, see [Hoffman and Johnson 2006a, Hoffman and Johnson 2006b, Hoffman 2005a], assuming sufficient regularity of $(\varphi^h, \theta^h)$:

$$|N(\sigma(u)) - N^h(\sigma(U))| \approx \sum_{K \in T} \mathcal{E}_{K,h},$$

(10)

where $\mathcal{E}_{K,h} = e_{D,h}^K + e_{M,h}^K$ is an error indicator for element $K$ in the mesh $T$, and

$$e_{D,h}^K = \frac{1}{|T|} \int_I \left( (|R_1(U, P)|_K + |R_2(U, P)|_K) \cdot (C_h h^2 |D^2 \varphi^h|_K + C_h k |\varphi^h_t|_K) 
+ \|R_3(U)\|_K \cdot (C_h h^2 \|D^2 \theta^h\|_K + C_h k \|\theta^h_t\|_K) \right) dt,$$

$$e_{M,h}^K = \frac{1}{|T|} \int_I SD_{\delta}(U, P; \varphi^h, \theta^h)_K dt,$$

where we may view $e_{D,h}^K$ as an error contribution from the Galerkin part of the cG(1)cG(1) discretization and $e_{M,h}^K$ as a contribution from the stabilization in cG(1)cG(1) on element $K$, and $k$ and $h$ are the time step size and the local mesh size, respectively. The residuals $R_i$ are defined by

$$R_1(U, P) = U_t + U \cdot \nabla U + \nabla P - f - \nu \Delta U,$$

$$R_2(U, P) = \nu D_2(U),$$

$$R_3(U, P) = \nabla \cdot U,$$

(11)
with
\[ D_2(U)(x, t) = \max_{y \in \partial K}(h_n(x))^{-1}\left|\left| \frac{\partial U}{\partial n}(y, t) \right|\right| \] (12)
for \( x \in K \), with \([\cdot]\) the jump across the element edge \( \partial K \). \( D^2 \) denotes second order spatial derivatives, and we write \( |w|_K \equiv (\|w_1\|_K, \|w_2\|_K, \|w_3\|_K) \), with \( \|w\|_K = (w, w)^{1/2}_K \), and let the dot denote the scalar product in \( \mathbb{R}^3 \).

In this paper, we keep the space mesh \( T \) and time step size \( k \) constant in time, with the time step being equal to the smallest element diameter in the space mesh. We use the following algorithm for adaptive mesh refinement in space, based on the a posteriori error estimate (10) for the approximation of mean drag:

Given an initial coarse computational space mesh \( T^0 \), start at \( k = 0 \), then do

1. Compute approximation of the primal problem using \( T^k \).
2. Compute approximation of the dual problem using \( T^k \).
3. If \( \sum_{K \in T^k} E_{K,h}^k < \text{TOL} \) then STOP, else:
4. Refine a fraction of the elements in \( T^k \) with largest \( E_{K,h}^k \rightarrow T^{k+1} \).
5. Set \( k = k + 1 \), then goto (1).

In the computations we use \( C_k = 1/2 \) and \( C_h = 1/8 \) as constant approximations of the interpolation constants in (10), where these values are motivated by simple analysis on reference elements. Boundary conditions other than no-slip, such as slip conditions at lateral boundaries and transparent outflow conditions [Hoffman and Johnson 2006a], introduce additional boundary terms in the a posteriori error estimate (10). But since the dual solutions for the problems in this paper are small at such boundaries, we neglect the corresponding boundary terms in (10).

12
In our computational model of a wheel we assume uniform rotation of a circular cylinder on flat ground, with the length of the cylinder being equal to its diameter $D = 0.1$. In a coordinate frame moving with the constant speed of the center axis of the cylinder, the problem is to determine the flow past a uniformly rotating circular cylinder with a fixed center axis and in contact with the ground moving with the same velocity as the oncoming free stream of the fluid. We set the kinematic viscosity $\nu = 10^{-5}$, resulting in a Reynolds number $Re = 10^4$ based on the free stream velocity $U_\infty = 1$ and the cylinder diameter $D = 0.1$.

The computational domain is set to a channel of dimension $[0, 30D] \times [0, 10D] \times [0, 20D]$, with the cylinder midpoint at $(5D, 0.5D, 10D)$. We use a constant unit streamwise inflow velocity $(U_\infty, 0, 0)$, slip boundary conditions on the spanwise boundaries, and a transparent outflow boundary condition [Hoffman and Johnson 2006a]. For the rotating cylinder the angular velocity is set to give a tangential velocity of magnitude $U_\infty$ on the curved surface of the cylinder, and the boundary conditions on the vertical boundaries of the channel are the same as the inflow condition.

For comparison we also study a model of a fixed wheel in a wind tunnel test. As a model we use a non rotating stationary cylinder where we change the boundary conditions to no slip on the cylinder and the vertical channel boundaries. In particular we note that the no slip boundary condition of the channel floor will introduce a boundary layer, which is not present for the rotating cylinder where the channel floor has the same velocity as the free stream in the frame of reference of the center axis of the cylinder.
8 Computational results

We now compare the computational results for the rotating cylinder to the results for the stationary cylinder. Of particular interest is qualitative and quantitative differences due to (i) the rotation of the cylinder and (ii) the artificial boundary layer for the stationary cylinder.

In Fig. 1–2 we plot snapshots of the velocity, pressure and vorticity for the 2 solutions. We note that one main difference between the two models is that for the stationary cylinder the flow separates from the channel floor to form a horse-shoe vortex upstream the cylinder, but for the rotating cylinder we have no separation upstream. The separation for the stationary cylinder is caused by the positive pressure gradient upstream in front of the cylinder and the artificial boundary layer (ii) at the channel floor. On the other hand, for the rotating cylinder we do not have a boundary layer and thus separation does not occur.

The presence of the horse-shoe vortex causes a reduction of the pressure on the upstream surface of the cylinder, leading to a reduced drag, and also influences the vorticity production. Near the line of contact for the cylinder a high amount of vorticity is produced for the rotating cylinder, whereas this production appears to be less significant for the stationary cylinder, probably due to the horse-shoe vortex leading to a redirection of the flow with low velocities close to the contact line leading to low vorticity.

We also find that the rotation (i) of the cylinder makes the flow separate earlier at the surface of the cylinder than in the case of the non-rotating cylinder, which makes the wake volume, and thus the drag, to increase.
The flows corresponding to the two cylinder configurations are thus fundamentally different; (i) the rotation of the cylinder causes earlier separation leading to an increase in wake volume, and thus drag, and (ii) the artificial boundary layer of the stationary cylinder results in a horse-shoe vortex upstream the cylinder leading to a reduced pressure on the upstream cylinder surface and a reduced vorticity production near the contact line of the cylinder. Both (i) and (ii) thus results in a higher drag for the rotating cylinder.

In our computations we use adaptive mesh refinement with respect to the error in drag for the two cylinder scenarios. We find that the resulting drag coefficients differs by almost a factor 2; for the rotating cylinder $c_D \approx 1.3$, to compare with $c_D \approx 0.8$ for the stationary cylinder, see Fig. 3–6.

We also note that the lift is higher for the rotating cylinder, which is expected since we have a higher pressure near the contact line for the rotating cylinder, without the horse-shoe vertex.

9 Reliability of computational results

The G2 method used in this paper is described in detail in [Hoffman and Johnson 2006b, Hoffman 2005a, Hoffman 2005b, Hoffman 2006a, Hoffman and Johnson 2006a], and we here only briefly comment on its main features: The mesh adaptivity is driven by minimizing the error in a user defined output, which here is chosen to be the drag. Thus the computational mesh in Fig. 9 is designed for an accurate computation of drag, which means that not necessarily other quantities are equally well approximated. Although, the experience from [Hoffman and Johnson 2006b, Hoffman 2005a, Hoffman 2005b, Hoffman 2006a, Hoffman and Johnson 2006a] in-
dicates that to accurately compute drag, the resulting approximation typically also has a high accuracy in other mean value quantities (such as lift, Strouhal numbers etc.).

In Fig. 3 we find that for the 4 finest meshes in the computations, drag is stable and we thus consider these approximations of drag to be credible. The dotted lines in Fig. 3–5 correspond to an alternative expression for drag where the stabilization term in equation (6) is dropped, and we find that for the finer meshes the two versions give the same results.

In Fig. 7–8 we plot the approximate dual solutions \((\varphi^h, \theta^h)\) for the rotating and the stationary cylinder, on the finest meshes respectively. The dual solutions act as a weight function for the residual in the a posteriori error estimates, and thus is a measure of how local errors (in the form of the residual) influence the error in output (here the drag).

We find that the overall structure of the two dual solutions are similar: the dual velocities \(\varphi^h\) are large in the boundary layer of the cylinder, in the turbulent wake, and upstream the cylinder. The dual pressures \(\theta^h\) are highest near the line of ground contact.

But there are also differences; High values of the dual velocity for the rotating cylinder is more spread out due to the larger wake, and for the stationary cylinder the high values of dual velocity convected upstream on top of the horse-shoe vortex, whereas for the rotating cylinder convection takes place along the channel floor.

In Fig. 9 we plot the refined meshes, where we note that for the stationary cylinder we need to capture the horse-shoe vortex, and for the rotating cylinder we need to refine the volume near the contact with the ground to a larger extent than for the
10 Summary

In this paper we have considered two computational models of a totally exposed wheel of a vehicle; the first model being a rotating cylinder on flat ground, which we compared to a second model of a stationary cylinder experiencing a free stream, with the purpose of simulating a simple wind tunnel test. For both models we considered turbulent flow at a Reynolds number of $10^4$.

There are two major differences between the two models: (i) the rotation of the cylinder, and (ii) the introduction of an artificial boundary layer at the channel floor for the stationary cylinder.

In computations using the G2 method we found that both (i) and (ii) qualitatively contribute to a higher drag for the rotating cylinder, which was also confirmed quantitatively in computations with a factor 2 difference in drag.

This large difference in drag have important practical implications; with a stationary car wind tunnel testing being an inadequate model of a rotating wheel.
References


Figure 1. Snapshots of velocity (upper) and pressure (lower); magnitude of velocity and pressure, and isosurfaces of negative pressure, for rotating (left) and stationary (right) cylinder, in the $x_1x_2$, $x_1x_3$, and $x_2x_3$-planes, through the center of the cylinder.
Figure 2. Snapshots of magnitude of the vorticity for the rotating (left) and the stationary (right) cylinder.
Figure 3. $\bar{c}_D^h$ vs $\log_{10}$ # nodes for the rotating (‘*’) and the stationary (‘o’) cylinder.
Figure 4. $c_L^{ch}$ vs $\log_{10}$ # nodes for the rotating (‘*’) and the stationary (‘o’) cylinder.
Figure 5. $c^h_2$ vs $\log_{10}$ # nodes for the rotating (‘*’) and the stationary (‘o’) cylinder.
Figure 6. Time series for $\bar{c}_D$ (‘.’) and $\bar{c}_L$, $\bar{c}_z$ (‘:’) for the rotating and the stationary cylinder, on the finest meshes respectively.
Figure 7. Snapshots of magnitude of the dual velocity for the rotating (left) and the stationary (right) cylinder, in the $x_1x_2$-plane (upper), and the $x_1x_3$-plane (lower), through the center of the cylinder.
Figure 8. Snapshots of the dual pressure for the rotating (left) and the stationary (right) cylinder, in the $x_1 x_2$-plane (upper), and the $x_1 x_3$-plane (lower), through the center of the cylinder.
Figure 9. Refined mesh for the rotating (left) and the stationary (right) cylinder, in the $x_1x_2^*$, $x_1x_3^*$, and $x_2x_3^*$-planes.