ECNS: a new dedicated Navier-Stokes solver for wind-turbine wake simulations

Benjamin Sanderse[1,2], **Margreet Nool**[2], **Steven van Haren**[4], **Barry Koren**[2,3]

[1] Energy research Centre of the Netherlands (ECN)
[2] Centrum Wiskunde & Informatica (CWI), The Netherlands
[3] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands
[4] Nuclear Research and consultancy Group (NRG), The Netherlands

E-mail: sanderse@ecn.nl

Abstract. In this article we describe a new Energy-Conserving Navier-Stokes code (ECNS) for the simulation of incompressible turbulent flow in wind-turbine wakes. The discretization in space and time is energy conserving, which results in two important properties: (i) the discretization does not introduce numerical viscosity and (ii) the discretization is stable for any mesh and any time step. A new class of non-diffusive actuator methods is used to simulate the effect of the wind turbine on the flow. The non-diffusive properties of the code are illustrated with the roll-up of a tip vortex and the flow through an actuator disk.

1. Introduction

In the last two decades wake calculations have progressed significantly: from 'parabolized' RANS (Reynolds-Averaged Navier-Stokes) simulations with turbines represented by actuator disks to 'elliptic' LES (Large Eddy Simulation) simulations with rotating actuator lines [1]. LES is clearly a more accurate approach than RANS due to the highly unsteady turbulent nature of wind turbine wakes [2]. However, LES computations of wakes require a different numerical treatment than RANS. Firstly, the 'turbulent' (modeled, eddy) viscosity in LES is much smaller than in RANS. In most RANS methods the eddy viscosity is so large that numerical viscosity is unnoticed, making results rather insensitive to details of the (convective) discretization. However, in LES methods numerical viscosity can easily overwhelm the physical viscosity and be detrimental for turbulence simulations [3, 4]. This requires a discretization with a minimum amount of numerical viscosity. Secondly, due to limitations in computer speed and memory, practical LES simulations of wind turbine wakes are mostly *under-resolved*. The order of (local) accuracy of a method, a notion related to the limit of vanishing mesh size, does not guarantee that (globally) accurate results will be obtained on coarse meshes. Instead, we will therefore focus on methods that possess some important properties on *coarse meshes* and for *large time steps*: well-posedness and stability. We aim to construct a spatial and temporal discretization that is ideally both *non-dissipative* (to prevent premature decay of turbulence), and *stable* (independent of the modeled viscosity and independent of the mesh or time step).

In order to achieve these properties, we will focus on so-called *energy-conserving* discretization methods, and propose a new Navier-Stokes solver for wind-turbine wake simulations: ECNS -

Energy-Conserving Navier-Stokes solver. Since we will use actuator methods to represent the effect of the turbines on the flow, we do not need to construct body-fitted grids. This is a great advantage since it allows us to use (staggered) cartesian grids, on which energy conservation is relatively straightforward to achieve. Staggered grids and the spatial discretization will be introduced in section 2. In section 3 a temporal discretization is proposed which keeps the energy conservation property when marching in time. The resulting (spatial and temporal) discretization is free of numerical viscosity and stable for any mesh and time step. In line with these properties we introduce in section 4 a new class of actuator methods that are free of the smearing effect present in conventional actuator methods. In section 5 details of the parallel implementation are discussed. Section 6 shows the energy-conserving properties and non-diffusive actuator treatment in a number of numerical experiments.

2. Spatial discretization

2.1. Governing equations

The governing equations for incompressible flow are those describing conservation of mass and momentum:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = -\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f}, \tag{2}$$

defined on a domain Ω and supplemented with suitable initial conditions and boundary conditions. The equation for kinetic energy $k = \frac{1}{2}|\boldsymbol{u}|^2$ is, for incompressible flows, derived from the momentum equation and as such expresses secondary conservation. Upon integration over the entire domain one obtains, with periodic or no-slip boundary conditions,

$$\frac{\mathrm{d}K}{\mathrm{d}t} = -\nu(\nabla \boldsymbol{u}, \nabla \boldsymbol{u}) + (\boldsymbol{f}, \boldsymbol{u}), \tag{3}$$

where the total kinetic energy is defined by $K = \int_\Omega k \,\mathrm{d}\Omega = \frac{1}{2}(\boldsymbol{u}, \boldsymbol{u})$. Equation (3) shows that the total energy of the flow is unaffected by the pressure and convective terms, but changed only by the diffusive terms and body forces. If equation (3) (with $\boldsymbol{f} = 0$) can be satisfied in a discrete sense, then the discrete solution is bounded, independent of grid layout or time step.

2.2. Staggered grid

We spatially discretize equations (1)-(2) using a finite volume method on a staggered cartesian grid, see figure 1, and [5, 6]. A staggered grid yields a natural coupling between the divergence operator in equation (1) and the gradient operator in equation (2). We obtain the following semi-discrete system of equations:

$$Mu(t) = 0, \tag{4}$$

$$\Omega\, \dot{u}(t) = -C(u(t)) + \nu D u(t) - Gp(t) + F(t). \tag{5}$$

The diagonal matrix $\Omega = \mathrm{diag}(\Omega^u, \Omega^v)$ contains the finite volume sizes of the u- and v-volumes, and M, C, G and D represent the discrete divergence, convection, gradient and diffusion operators, and F the effect of body forces. By approximating all integrals with the midpoint method we obtain the following properties:

$$(Gp, u) = -(p, Mu) = 0, \tag{6}$$

$$(C(u), u) = 0, \tag{7}$$

$$(Du, v) = (u, Dv), \quad (Du, u) \leq 0. \tag{8}$$

The convective terms do not contribute to the energy equation (due to central differencing), and the pressure terms do not contribute due to the strong coupling between velocity and pressure, inherent to the staggered grid approach. This gives the following semi-discrete equivalent of equation (3):

$$\frac{\mathrm{d}K}{\mathrm{d}t} = \nu(Du, u). \tag{9}$$

In the limit of vanishing viscosity the discretization ensures that energy is conserved in space.

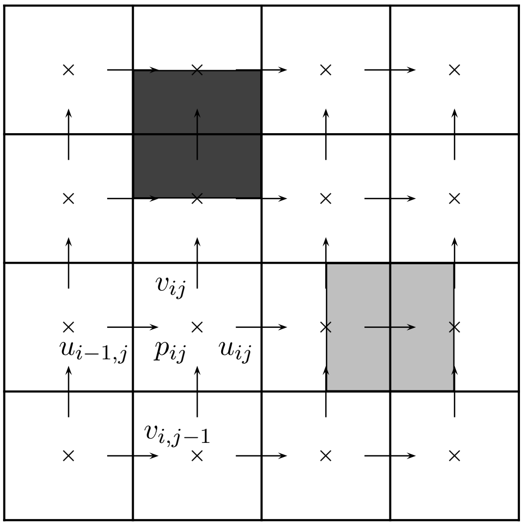


Figure 1: Staggered grid example with p-volumes (white) and an example of a u-volume (light gray) and v-volume (dark gray).

3. Temporal discretization

We consider Runge-Kutta methods for time integration of equations (4)-(5). Runge-Kutta methods can combine high-order with good stability and efficiency, they allow easy adaptive step size selection and are self-starting. A general (implicit or explicit) Runge-Kutta method applied to equations (4)-(5) reads [7, 8]:

$$U_i = u_n + \Delta t \sum_{j=1}^{s} a_{ij}(N_j - Gp_j), \qquad MU_i = 0, \tag{10}$$

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^{s} b_i(N_i - Gp_{n+1}), \qquad Mu_{n+1} = 0, \tag{11}$$

with a_{ij}, b_i, c_i the weights of the Runge-Kutta method, $t_i = t_n + c_i \Delta t$, s the number of stages, and N containing convective and diffusive terms. We will choose the weights of the Runge-Kutta method such that energy is also conserved in time. Upon taking the square of equation (11) and inserting (10) we obtain the fully discrete evolution equation for the kinetic energy:

$$\|u_{n+1}\|^2 = \|u_n\|^2 + 2\Delta t \sum_{i=1}^{s} b_i \left(PN_i, U_i\right) - \Delta t^2 \sum_{i,j=1}^{s} \left(b_i a_{ij} + b_j a_{ji} - b_i b_j\right) \left(PN_i, PN_j\right). \tag{12}$$

Here we have introduced for convenience the projection operator $P = I - GL^{-1}M$ which ensures that the solution stays divergence-free. Conservation of energy requires that the last two terms in (12) vanish. The term (PN_i, U_i), is zero due to our spatial discretization: $(C(U_i), U_i) = 0$ and $(Gp_i, U_i) = 0$. For the last term to be zero the coefficients a and b from the Runge-Kutta tableau should satisfy

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0. \tag{13}$$

A class of methods that satisfies this condition is based on Gauss quadrature, see table 1. The lowest (second order) method is also known as the *implicit midpoint method.* With the Gauss methods, the fully discrete energy equation (12) can be written as

$$\frac{\frac{1}{2}\|u_{n+1}\|^2 - \frac{1}{2}\|u_n\|^2}{\Delta t} = \sum_{i=1}^{s} b_i (DU_i, U_i), \tag{14}$$

which is the discrete equivalent of equation (3). The discrete energy can therefore, like the continuous energy, only decrease, due to viscous effects. In case viscosity is absent the energy stays exactly constant in time. The discretization is therefore stable and non-dissipative.

We note that the choice for implicit or explicit time integration methods depends on the problem at hand. If the time step of the problem is dictated by accuracy rather than stability it might be better to opt for explicit methods, even though this introduces an energy conservation error. In that case we use one of the explicit Runge-Kutta methods mentioned in [7].

Table 1: Gauss methods, 2nd and 4th order.

$\frac{1}{2}$	$\frac{1}{2}$
	1

$\frac{1}{2} - \frac{\sqrt{3}}{6}$	$\frac{1}{4}$	$\frac{1}{4} - \frac{\sqrt{3}}{6}$
$\frac{1}{2} + \frac{\sqrt{3}}{6}$	$\frac{1}{4} + \frac{\sqrt{3}}{6}$	$\frac{1}{4}$
	$\frac{1}{2}$	$\frac{1}{2}$

4. Actuator methods

In sections 2 and 3 we have introduced a discretization in space and time that does not introduce artificial diffusion and is stable for any mesh and any time step. We will pursue here this non-dissipative approach when modeling actuators.

Actuator methods are used in wind-turbine wake simulations to represent the action of the blades on the flow without the need of constructing a (body-fitted) computational grid. A widely used method is based on discrete Dirac functions, which spread the force on the actuator to the surrounding grid points, see e.g. [9]. Such methods have a free 'regularization' parameter and introduce some 'smearing' depending on this parameter. We will consider a new class of methods, Immersed Actuator Methods (IAM), that do no have such a free parameter and which introduce less smearing effects. Here we give a short introduction into the IAM - for more information we refer to [10] (published in this issue).

4.1. Steady actuators

We consider the incompressible Navier-Stokes equations with an external force $\boldsymbol{F}$ in *integral form*, since we will be dealing with discontinuities:

$$\int_{\Omega} \frac{\partial \boldsymbol{u}}{\partial t} \, \mathrm{d}\Omega + \int_{\Gamma} \boldsymbol{u}\,\boldsymbol{u} \cdot \boldsymbol{n} \, \mathrm{d}\Gamma = \int_{\Gamma} -p\,\boldsymbol{n} \, \mathrm{d}\Gamma + \int_{\Gamma} \nu(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T) \cdot \boldsymbol{n} \, \mathrm{d}\Gamma + \boldsymbol{F}, \tag{15}$$

where Ω is a control volume with boundary Γ, see figure 2a. In actuator methods $\boldsymbol{F}$ represents the action of a body $\mathcal{S}$ on the flow and can be written as a surface integral of pressure and shear stresses:

$$\boldsymbol{F} = \int_{\mathcal{S}} \boldsymbol{f}^b(\boldsymbol{\xi}(s))\,\mathrm{d}A \left(= \int_{\mathcal{S}} \boldsymbol{f}^b(\boldsymbol{\xi}(s)) \left|\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}s}\right| \mathrm{d}s \right), \tag{16}$$

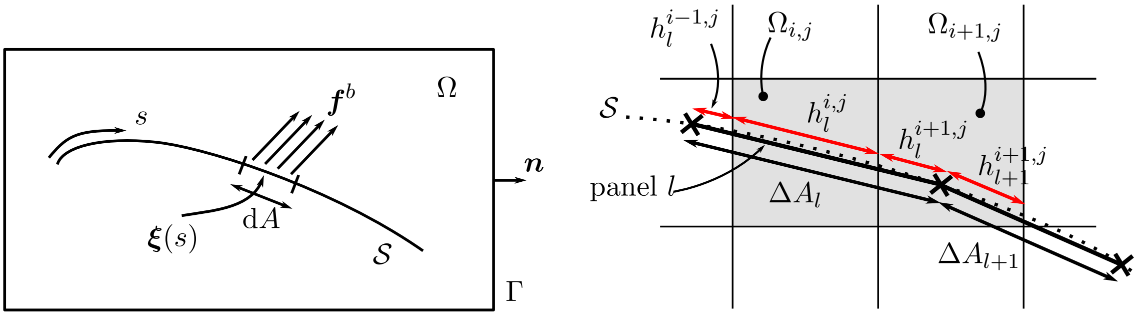
where $\boldsymbol{f}^b$ is a surface *stress* (force per unit area) which depends on the position on the surface $\boldsymbol{\xi}$ parameterized by s; see figure 2a. We discretize an actuator surface by representing it by N piecewise linear segments ('panels'). The integral (16) changes into a summation over these panels:

$$\boldsymbol{F} = \sum_{l=1}^{N} \boldsymbol{f}_l^b \Delta A_l, \tag{17}$$

where ΔA_l is the area of panel l and $\boldsymbol{f}_l^b$ the surface stress acting on it. With the actuator discretized in panels, we apply equation (15) to a finite volume. Like in the continuous case, a volume $\Omega_{i,j}$ gets a contribution from $\boldsymbol{f}_l^b$ if $\Omega_{i,j}$ is intersected by the body $\mathcal{S}$ (see figure 2b). The force on finite volume $\Omega_{i,j}$ is simply given by

$$\boldsymbol{F}_{i,j} = \sum_{l=1}^{N} \boldsymbol{f}_l^b h_l^{i,j}, \tag{18}$$

where $h_l^{i,j}$ is the area of panel l contained in volume $\Omega_{i,j}$.



(a) Continuous case

(b) Discrete case

Figure 2: Steady actuator with surface forces.

4.2. Moving actuators

In the unsteady case, equation (15) is integrated in both space and time. Assuming a stationary volume Ω we obtain

$$\left[\int_{\Omega} \boldsymbol{u}\,\mathrm{d}\Omega\right]_{t_1}^{t_2} = \int_{t_1}^{t_2} \left(\int_{\Gamma} \left(-\boldsymbol{u}\boldsymbol{u} + \nu(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T)\right) \cdot \boldsymbol{n} - p\,\boldsymbol{n}\,\mathrm{d}\Gamma \right) \mathrm{d}t + \int_{t_1}^{t_2} \boldsymbol{F}(t)\,\mathrm{d}t. \tag{19}$$

We consider an actuator surface $\mathcal{S}$ moving in time. In two dimensions the actuator surface is a contour which traces out an area V in time. This area is parameterized as $\boldsymbol{\xi}(s,t)$, see figure 3a. The total impulse exerted by the actuator on the flow is given by

$$\boldsymbol{I} = \int_{t_1}^{t_2} \boldsymbol{F}(t)\,\mathrm{d}t = \int_{t_1}^{t_2} \int_{\mathcal{S}} \boldsymbol{f}^b(\boldsymbol{\xi}(s,t),t)\,\mathrm{d}A\,\mathrm{d}t = \int_{t_1}^{t_2} \int_{\mathcal{S}} \boldsymbol{f}^b(\boldsymbol{\xi}(s,t),t) \left|\frac{\partial\boldsymbol{\xi}}{\partial s}\right| \mathrm{d}s\,\mathrm{d}t. \tag{20}$$

The integral in terms of the parametric representation (s,t) can be written in terms of an integral in space. For details we refer to [10]. The resulting equation is

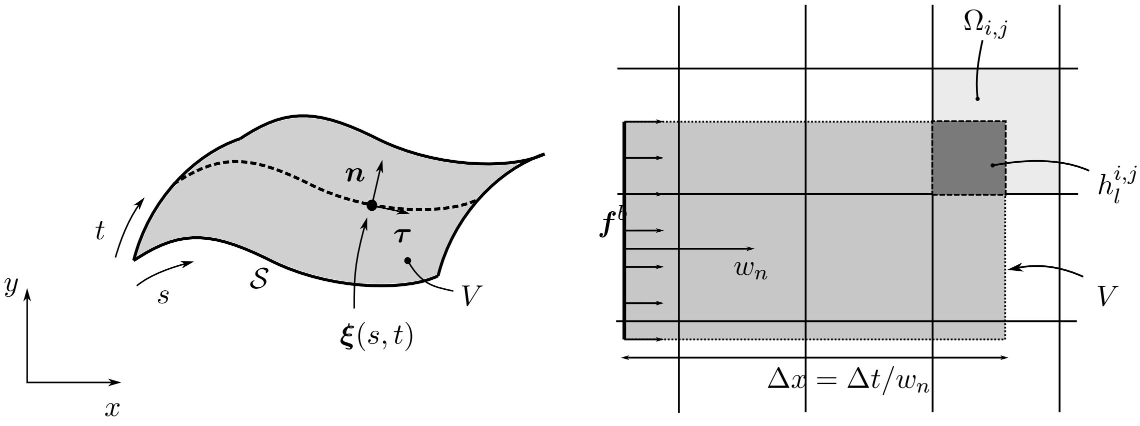
$$\boldsymbol{I} = \iint\limits_{V} \frac{\boldsymbol{f}^b(\boldsymbol{x})}{|w_n(\boldsymbol{x})|}\,\mathrm{d}V, \tag{21}$$

where w_n is the velocity of the actuator normal to its surface. In this way the integral in time has been transformed into an integral in space, which leads to a form more suitable for discretization.

For illustration purposes we assume a force uniform in space and an actuator moving with constant velocity in time, see figure 3b. A finite volume $\Omega_{i,j}$ gets a contribution if it is intersected by the area swept out by the moving actuator:

$$\boldsymbol{I}_{i,j} = \left(\int_{t^n}^{t^{n+1}} \boldsymbol{F}(t)\,\mathrm{d}t\right)_{i,j} = \frac{\boldsymbol{f}^b}{|w_n|}\iint\limits_{V\cap\Omega_{i,j}} \mathrm{d}V = h_l^{i,j}\frac{\boldsymbol{f}^b}{|w_n|}, \tag{22}$$

where $h_l^{i,j}$ is the part of $\Omega_{i,j}$ swept out by panel l during a time step. The advantage is that this discretization conserves the total impulse that is added to the flow. Furthermore, our proposed discretization is suitable for large time steps: it exerts a force over the entire region where the actuator has passed.



(a) Continuous case

(b) Discrete case

Figure 3: Actuator surface moving in time.

5. Parallelization

The ECNS code is parallelized with the Portable Extensible Toolkit for Scientific Computation (PETSc), ‘a suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations’ [11]. We build our discretization in terms of sparse PETSc matrices and vectors. Once constructed, the user can perform matrix-vector manipulations (such as matrix-vector and matrix-matrix multiplications, dot products, etc.) without being concerned about the specifics of the parallel implementation. Since all our operators (convection, diffusion, etc.) are programmed as matrices, the implementation of implicit time discretizations is relatively straightforward. Furthermore a large collection of linear algebra solvers is available within PETSc, and many existing external solver libraries can be linked to it. For example, for the solution of the Poisson equation for the pressure we are using the conjugate gradient method from PETSc with algebraic multigrid as preconditioner from the external package ML [12].

6. Results

In this section we present two numerical experiments that support the work of the previous sections. The first shows the energy conservation properties in space and time of our discretization, and the second shows the sharp, non-diffusive representation of actuators immersed in a flow.

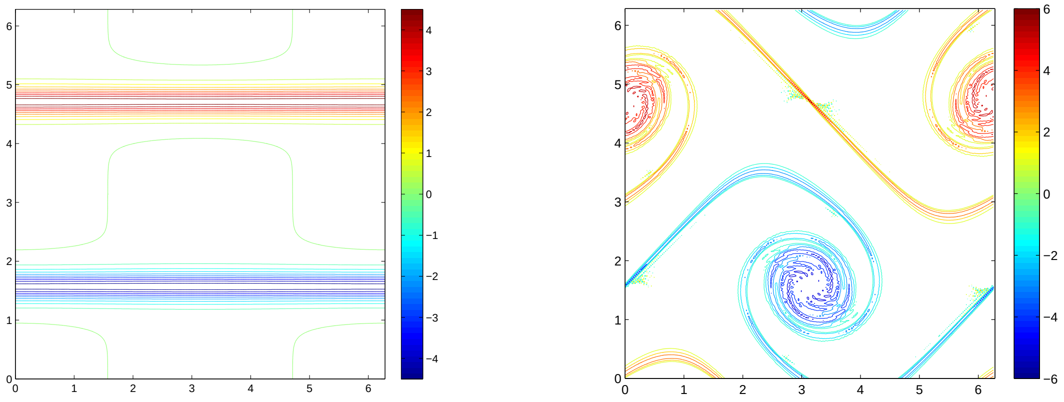
6.1. Energy conservation

In order to investigate the energy conservation properties of ECNS we study the roll-up of a vortex originating from the tip of a rotating wind turbine blade. We take an idealized setting: a periodic domain of dimensions $[0, 2\pi] \times [0, 2\pi]$ with initial condition [13, 14]:

$$u = \begin{cases} \tanh\left(\frac{y-\pi/2}{\delta}\right), & y \leq \pi, \\ \tanh\left(\frac{3\pi/2-y}{\delta}\right), & y > \pi, \end{cases} \qquad v = \varepsilon \sin(x), \tag{23}$$

with $\delta = \pi/15$, $\varepsilon = 0.05$. The viscosity of the flow is set to zero, but the perturbation of the v-velocity will make sure that the shear-layer starts to roll-up. Impressions of the flow field at $t = 0$ and $t = 8$ are given in figure 4. Since the flow is inviscid, smaller and smaller scales will be created by the non-linear convective term, so that at some point the mesh is too coarse to capture the solution: noise appears in the solution in the form of wiggles. The flow field is then under-resolved, something which generally occurs in simulations of wind-turbine wakes.

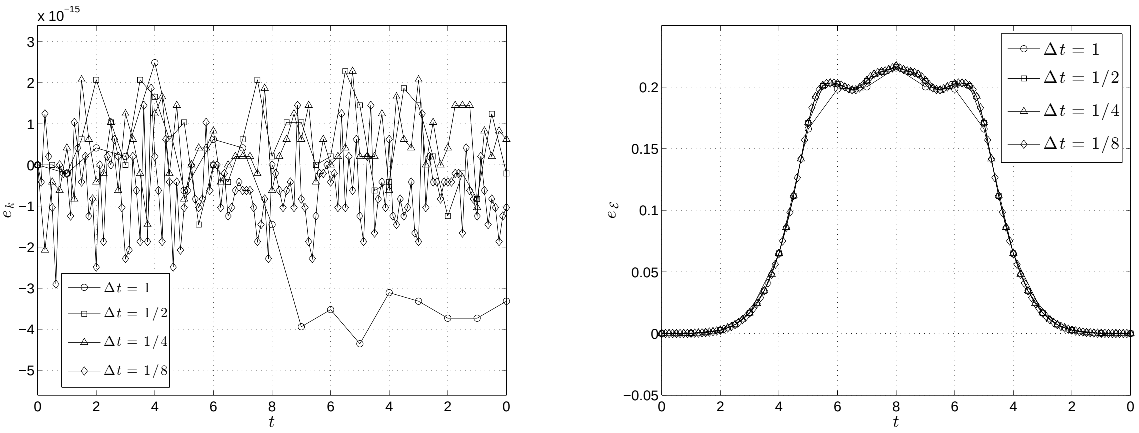
Figure 5a shows a typical behavior of the energy error $(k(t) - k(t = 0))$ as a function of time during the vortex roll-up. It can be seen that the error stays at machine precision, independent of the time step. The simulation does not blow-up, even if the flow field becomes more and more under-resolved. Figure 5b shows the enstrophy error (enstrophy is not conserved by the spatial discretization), which indicates that the simulation is not only energy-conserving, but also time-reversible. For more details we refer to [8].



(a) $t = 0$

(b) $t = 8$

Figure 4: Vorticity field of shear layer roll-up at different time instances, computed with a uniform grid with 256^2 volumes. Iso-contours $-4 \ldots 4$ with steps 0.5.



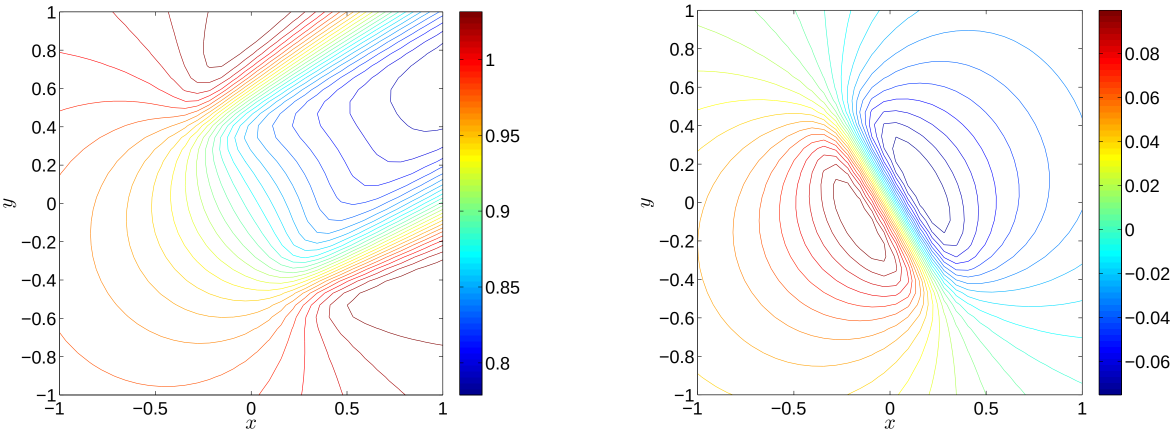
(a) Energy (showing round-off errors)

(b) Enstrophy

Figure 5: Energy and enstrophy errors for Gauss 4 applied to shear-layer problem.

6.2. Actuator disk simulation

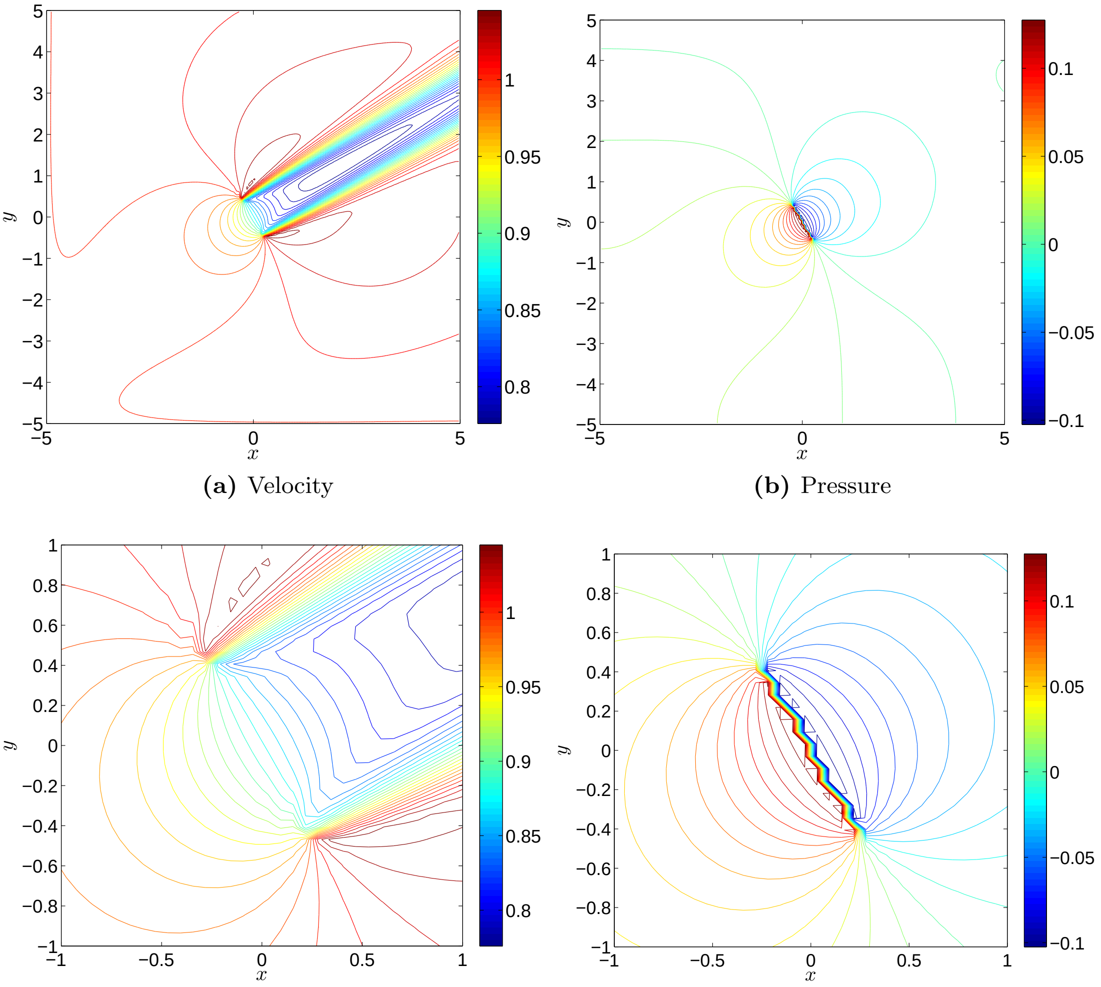
We will now consider the flow through an actuator disk at an angle of 30 degrees. A uniformly loaded actuator disk presents a challenging test case due to the presence of a singularity at the disk edge, where high gradients appear and vorticity is created. We take $C_T = \frac{1}{2}$ and $\text{Re} = 100$. For studies on the effect of Reynolds number, thrust coefficient and mesh resolution we refer to [15]. Figures 7a and 7b show the velocity and pressure in the entire flow domain. Figures 7c and 7d show a detail of the solution near the actuator disk. Compared to the conventional discrete Dirac approach, shown in figures 6a - 6b, we see that both pressure and velocity contours are more sharply captured. The small wiggles in the velocity contours of figure 7c reveal ('warn') that the mesh is too coarse (the mesh Péclet condition is not satisfied), whereas this warning is absent in figure 6a. The smearing effect of the discrete Dirac approach also leads to lower pressure values at both sides of the jump; this can lead to underestimation of power and blade loading when the method is applied to rotating turbine blades.



(a) Velocity

(b) Pressure

Figure 6: Flow through a tilted actuator disk, discrete Dirac approach: Gaussian with $\varepsilon = 2\Delta x$.



(a) Velocity

(b) Pressure

(c) Velocity, detail

(d) Pressure, detail

Figure 7: Flow through a tilted actuator disk, immersed actuator method.

6.3. Flow over a wing

To show that the proposed actuator method also work in 3D, we present the laminar flow over a uniformly loaded rectangular wing, figure 8, computed with the parallelized version of the code. The streamlines show the downwash behind the wing, and the formation of tip vortices. In future work we will apply the moving actuator method from section 4.2 to this wing, leading to a model for a rotating wind turbine blade.

7. Conclusions and future work

In this article we have presented the ingredients of an Energy-Conserving Navier-Stokes solver (ECNS): (i) a spatial discretization on a staggered grid, (ii) a temporal discretization with Gauss Runge-Kutta methods and (iii) immersed actuator methods to represent wind turbines. The common rationale behind these ingredients is to obtain a numerically non-dissipative, yet stable discretization, able to perform turbulence simulations with LES on coarse grids.

Ongoing work is on the selection of a proper LES model to represent the behavior of the smallest unresolved scales. The basic idea behind LES is that only the large energy-containing scales of the flow are resolved (scales larger than some predefined length scale Δ). An important

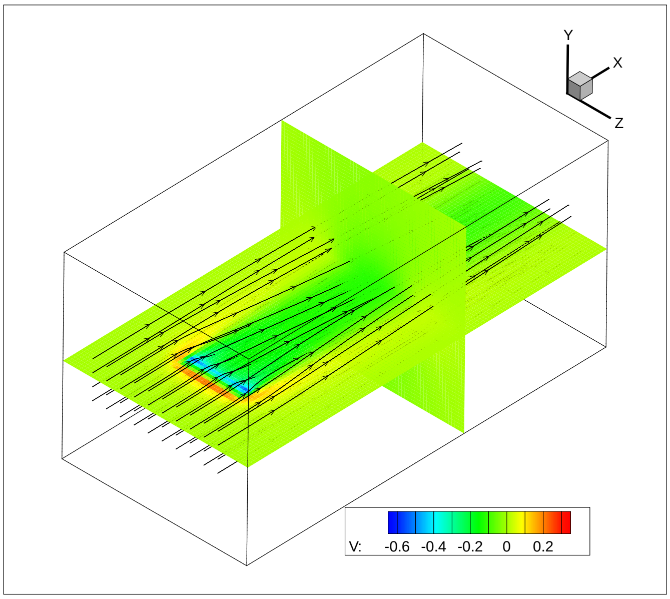


Figure 8: Flow over an actuator surface representing a wing. Vertical velocity contours.

question, which is in line with our non-dissipative discretization approach, is what the *minimum* required eddy viscosity is to damp scales smaller than Δ sufficiently. This question is addressed in [16]. It turns out that the eddy viscosity should not only depend on the magnitude of the strain-rate tensor, $\|S\|$, (as is the case in classical eddy-viscosity models), but also on its second invariant, namely its determinant. The resulting qr-model has some interesting properties, for example: the eddy viscosity vanishes in laminar flow, near walls and when Δ is in the Kolmogorov scale. This model will be the subject of future work.

Acknowledgments

The parallelization work, performed by Margreet Nool and Steven van Haren, was funded through an NCF Research Grant, grant number NRG-2011.06.

References

[1] Sanderse B, van der Pijl S and Koren B 2011 *Wind Energ.* **14** 799–819
[2] Réthoré P E 2009 *Wind turbine wake in atmospheric turbulence* Ph.D. thesis Aalborg University
[3] Mittal R and Moin P 1997 *AIAA Journal* **35**
[4] Mahesh K, Constantinescu G and Moin P 2004 *J. Comput. Phys.* **197** 215–240
[5] Harlow F and Welch J 1965 *Phys. Fluids* **8** 2182–9
[6] Verstappen R and Veldman A 2003 *J. Comput. Phys.* **187** 343–368
[7] Sanderse B and Koren B 2012 *J. Comput. Phys.* **231** 3041–3063
[8] Sanderse B 2012 *J. Comput. Phys.* Doi:10.1016/j.jcp.2012.07.039
[9] Sørensen J and Shen W 2002 *J. of Fluid. Eng.* **124** 393–399
[10] Sanderse B and Koren B 2013 *J. Phys.: Conf. Ser.*
[11] Balay S et al 2012 PETSc Users Manual Tech. Rep. ANL-95/11 - Revision 3.3 Argonne National Laboratory
[12] Gee MW et al 2006 ML 5.0 Smoothed Aggregation User's Guide Tech. Rep. SAND2006-2649 Sandia National Laboratories
[13] Brown D and Minion M 1997 *J. Comput. Phys.* **138** 734–765
[14] Knikker R 2009 *Int. J. Numer. Meth. Fluids* **59** 1063–1092
[15] Sanderse B 2011 ECNS: Energy-Conserving Navier-Stokes Solver. Verification of steady laminar flows Tech. Rep. ECN-E–11-042 Energy research Centre of the Netherlands
[16] Verstappen R 2011 *J. Sci. Comput.* **49** 94–110