Application of lattice Boltzmann methods for large-eddy simulation of wind turbine rotor wake aerodynamics

Ralf Deiterding, Christos Gkoudesnes, Aden Cox

Aerodynamics and Flight Mechanics Research Group
University of Southampton
Highfield Campus, Southampton SO17 1BJ, UK
E-mail: r.deiterding@soton.ac.uk

June 14, 2018

Outline

Approximation of Boltzmann equation

Is based on solving the Boltzmann equation with a simplified collision operator

$$\partial_t f + \mathbf{u} \cdot \nabla f = \omega(f^{eq} - f) + F$$

- $\mathrm{Kn} = l_f / L \ll 1$, where l_f is replaced with Δx
- Weak compressibilty and small Mach number assumed

Approximation of Boltzmann equation

Is based on solving the Boltzmann equation with a simplified collision operator

$$\partial_t f + \mathbf{u} \cdot \nabla f = \omega(f^{eq} - f) + F$$

- $\mathrm{Kn} = l_f / L \ll 1$, where l_f is replaced with Δx
- Weak compressibilty and small Mach number assumed

Equation is approximated in simplified phase space and with a splitting approach.

1.) Transport step solves $\partial_t f_\alpha + \mathbf{e}_\alpha \cdot \nabla f_\alpha = 0$

Operator: $\mathcal{T}$: $\tilde{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t)$

$$\rho(\mathbf{x}, t) = \sum_{\alpha=0}^{18} f_\alpha(\mathbf{x}, t), \quad \rho(\mathbf{x}, t) u_i(\mathbf{x}, t) = \sum_{\alpha=0}^{18} \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x}, t)$$

Approximation of Boltzmann equation

Is based on solving the Boltzmann equation with a simplified collision operator

$$\partial_t f + \mathbf{u} \cdot \nabla f = \omega(f^{eq} - f) + F$$

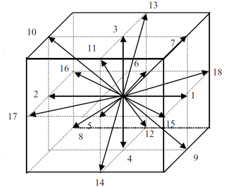
- $\mathrm{Kn} = l_f / L \ll 1$, where l_f is replaced with Δx
- Weak compressibilty and small Mach number assumed

Equation is approximated in simplified phase space and with a splitting approach.

1.) Transport step solves $\partial_t f_\alpha + \mathbf{e}_\alpha \cdot \nabla f_\alpha = 0$

Operator: $\mathcal{T}$: $\tilde{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t)$

$$\rho(\mathbf{x}, t) = \sum_{\alpha=0}^{18} f_\alpha(\mathbf{x}, t), \quad \rho(\mathbf{x}, t) u_i(\mathbf{x}, t) = \sum_{\alpha=0}^{18} \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x}, t)$$



Discrete velocities:

$$\mathbf{e}_\alpha = \begin{cases} 0, & \alpha = 0, \\ (\pm1, 0, 0)c, (0, \pm1, 0)c, (0, 0, \pm1)c, & \alpha = 1, \ldots, 6, \\ (\pm1, \pm1, 0)c, (\pm1, 0, \pm1)c, (0, \pm1, \pm1)c, & \alpha = 7, \ldots, 18 \end{cases}$$

$c = \dfrac{\Delta x}{\Delta t}$, Physical speed of sound: $c_s = \dfrac{c}{\sqrt{3}}$

Approximation of thermal equilibrium

2.) Collision step solves $\partial_t f_\alpha = \omega(f_\alpha^{eq} - f_\alpha) + F_\alpha$
Operator $\mathcal{C}$:

$$f_\alpha(\cdot, t+\Delta t) = \tilde{f}_\alpha(\cdot, t+\Delta t) + \omega_L \Delta t \left(\tilde{f}_\alpha^{eq}(\cdot, t+\Delta t) - \tilde{f}_\alpha(\cdot, t+\Delta t)\right) + \Delta t F_\alpha$$

with $F_\alpha = 3\rho t_\alpha \mathbf{e}_\alpha \mathbf{F}/c^2$ and equilibrium function

$$f_\alpha^{eq}(\rho, \mathbf{u}) = \rho t_\alpha \left[1 + \frac{3\mathbf{e}_\alpha \mathbf{u}}{c^2} + \frac{9(\mathbf{e}_\alpha \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} + \qquad \right]$$

with $t_\alpha = \frac{1}{9}\left\{3, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \right\}$

Approximation of thermal equilibrium

2.) Collision step solves $\partial_t f_\alpha = \omega(f_\alpha^{eq} - f_\alpha) + F_\alpha$
Operator $\mathcal{C}$:

$$f_\alpha(\cdot, t+\Delta t) = \tilde{f}_\alpha(\cdot, t+\Delta t) + \omega_L \Delta t \left(\tilde{f}_\alpha^{eq}(\cdot, t+\Delta t) - \tilde{f}_\alpha(\cdot, t+\Delta t)\right) + \Delta t F_\alpha$$

with $F_\alpha = 3\rho t_\alpha \mathbf{e}_\alpha \mathbf{F}/c^2$ and equilibrium function

$$f_\alpha^{eq}(\rho, \mathbf{u}) = \rho t_\alpha \left[1 + \frac{3\mathbf{e}_\alpha \mathbf{u}}{c^2} + \frac{9(\mathbf{e}_\alpha \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} + \frac{\mathbf{e}_\alpha \mathbf{u}}{3c^2}\left(\frac{9(\mathbf{e}_\alpha \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2}\right)\right]$$

with $t_\alpha = \frac{1}{9}\left\{3, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \right\}$

Approximation of thermal equilibrium

2.) Collision step solves $\partial_t f_\alpha = \omega(f_\alpha^{eq} - f_\alpha) + F_\alpha$
Operator $\mathcal{C}$:

$$f_\alpha(\cdot, t+\Delta t) = \tilde{f}_\alpha(\cdot, t+\Delta t) + \omega_L \Delta t \left(\tilde{f}_\alpha^{eq}(\cdot, t+\Delta t) - \tilde{f}_\alpha(\cdot, t+\Delta t) \right) + \Delta t F_\alpha$$

with $F_\alpha = 3\rho t_\alpha \mathbf{e}_\alpha \mathbf{F}/c^2$ and equilibrium function

$$f_\alpha^{eq}(\rho, \mathbf{u}) = \rho t_\alpha \left[1 + \frac{3\mathbf{e}_\alpha \mathbf{u}}{c^2} + \frac{9(\mathbf{e}_\alpha \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} + \frac{\mathbf{e}_\alpha \mathbf{u}}{3c^2} \left(\frac{9(\mathbf{e}_\alpha \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} \right) \right]$$

with $t_\alpha = \frac{1}{9}\left\{3, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \right\}$
Pressure $\delta p = \sum_\alpha f_\alpha^{eq} c_s^2 = \rho c_s^2$.
Dev. stress $\Sigma_{ij} = \left(1 - \dfrac{\omega_L \Delta t}{2}\right) \sum_\alpha \mathbf{e}_{\alpha i} \mathbf{e}_{\alpha j} (f_\alpha^{eq} - f_\alpha)$

Approximation of thermal equilibrium

2.) Collision step solves $\partial_t f_\alpha = \omega(f_\alpha^{eq} - f_\alpha) + F_\alpha$
Operator $\mathcal{C}$:

$$f_\alpha(\cdot, t+\Delta t) = \tilde{f}_\alpha(\cdot, t+\Delta t) + \omega_L \Delta t \left(\tilde{f}_\alpha^{eq}(\cdot, t+\Delta t) - \tilde{f}_\alpha(\cdot, t+\Delta t) \right) + \Delta t F_\alpha$$

with $F_\alpha = 3\rho t_\alpha \mathbf{e}_\alpha \mathbf{F}/c^2$ and equilibrium function

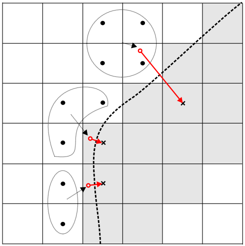
$$f_\alpha^{eq}(\rho, \mathbf{u}) = \rho t_\alpha \left[1 + \frac{3\mathbf{e}_\alpha \mathbf{u}}{c^2} + \frac{9(\mathbf{e}_\alpha \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} + \frac{\mathbf{e}_\alpha \mathbf{u}}{3c^2} \left(\frac{9(\mathbf{e}_\alpha \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} \right) \right]$$

with $t_\alpha = \frac{1}{9}\left\{3, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \right\}$
Pressure $\delta p = \sum_\alpha f_\alpha^{eq} c_s^2 = \rho c_s^2$.
Dev. stress $\Sigma_{ij} = \left(1 - \frac{\omega_L \Delta t}{2}\right) \sum_\alpha \mathbf{e}_{\alpha i} \mathbf{e}_{\alpha j} (f_\alpha^{eq} - f_\alpha)$
A Chapman-Enskog expansion ($f_\alpha = f_\alpha(0) + \epsilon f_\alpha(1) + \epsilon^2 f_\alpha(2) + \ldots$) shows that

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \qquad \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F}$$

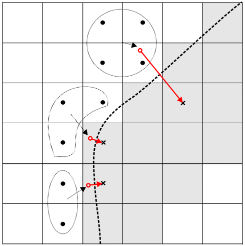
are recoverd to $O(\epsilon^{2,3})$ [Hou et al., 1996] and also $\omega_L = \tau_L^{-1} = \dfrac{c_s^2}{\nu + \Delta t c_s^2/2}$

Level-set method for boundary embedding



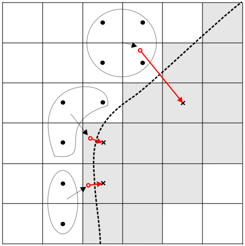
- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$.
- Construction of macro-values in embedded boundary cells by interpolation / extrapolation.
- Complex boundary moving with local velocity $\mathbf{w}$, ghost cell velocity: $\mathbf{u}' = 2\mathbf{w} - \mathbf{u}$

Level-set method for boundary embedding



- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$.
- Construction of macro-values in embedded boundary cells by interpolation / extrapolation.
- Complex boundary moving with local velocity $\mathbf{w}$, ghost cell velocity: $\mathbf{u}' = 2\mathbf{w} - \mathbf{u}$
- Then use $f_\alpha^{eq}(\rho', \mathbf{u}')$ or use interpolated bounce-back [Bouzidi et al., 2001] to construct distributions in embedded ghost cells.
- Distance computation for triangulated grids with CPT algorithm [Mauch, 2000].

Level-set method for boundary embedding



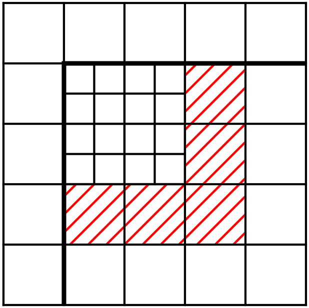
- Implicit boundary representation via distance function φ, normal $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$.
- Construction of macro-values in embedded boundary cells by interpolation / extrapolation.
- Complex boundary moving with local velocity $\mathbf{w}$, ghost cell velocity: $\mathbf{u}' = 2\mathbf{w} - \mathbf{u}$
- Then use $f_\alpha^{eq}(\rho', \mathbf{u}')$ or use interpolated bounce-back [Bouzidi et al., 2001] to construct distributions in embedded ghost cells.
- Distance computation for triangulated grids with CPT algorithm [Mauch, 2000].

Block-structured adaptive mesh refinement (SAMR)

- Refinement in all spatial directions and time by same factor
- Refined blocks overlay coarser ones
- Most efficient LBM implementation with patch-wise for-loops
- LBM implemented on finite volume grids
- AMROC V3.0 with significantly enhanced parallelization [Deiterding and Wood, 2015, Deiterding, 2011, Deiterding et al., 2007, Deiterding et al., 2006]

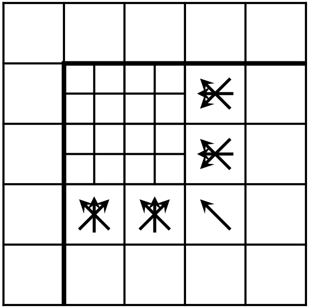
Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$



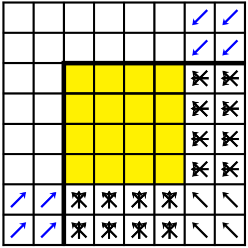
Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.



Adaptive LBM

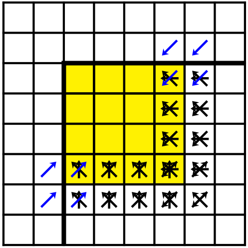
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.



$f_{\alpha,in}^{f,n}$

Adaptive LBM

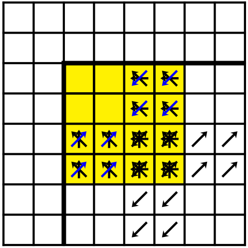
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.



$\tilde{f}_{\alpha,in}^{f,n}$

Adaptive LBM

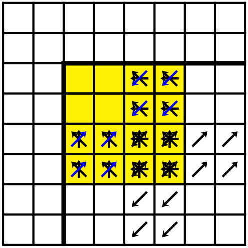
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



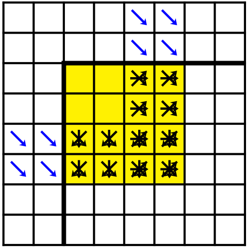
$\tilde{f}_{\alpha,in}^{f,n+1/2}$

Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



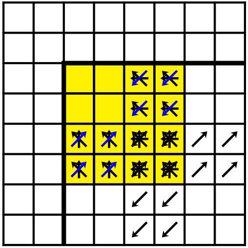
$\tilde{f}_{\alpha,in}^{f,n+1/2}$



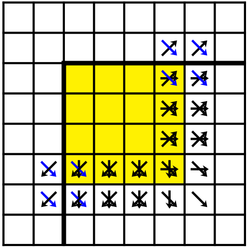
$f_{\alpha,out}^{f,n}$

Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



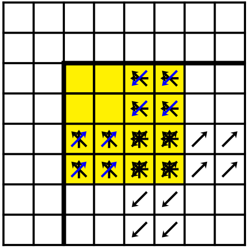
$\tilde{f}_{\alpha,in}^{f,n+1/2}$



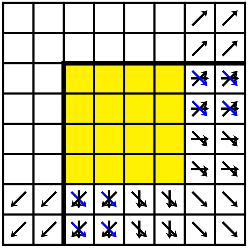
$\tilde{f}_{\alpha,out}^{f,n}$

Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



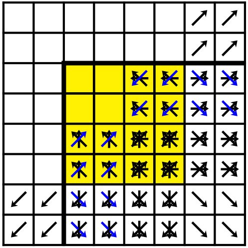
$\tilde{f}_{\alpha,in}^{f,n+1/2}$



$\tilde{f}_{\alpha,out}^{f,n+1/2}$

Adaptive LBM

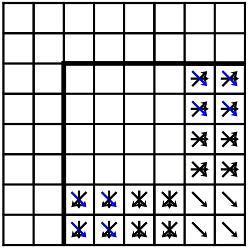
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



$\tilde{f}_{\alpha,out}^{f,n+1/2}$, $\tilde{f}_{\alpha,in}^{f,n+1/2}$

Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.

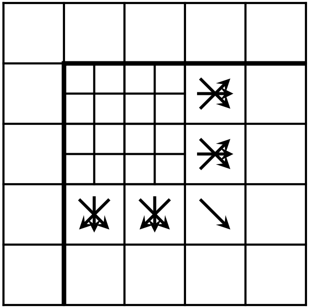


$\tilde{f}_{\alpha,out}^{f,n+1/2}$, $\tilde{f}_{\alpha,out}^{f,n}$

5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.

Adaptive LBM

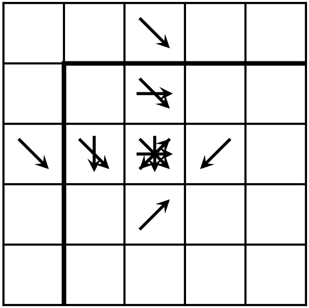
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.

Adaptive LBM

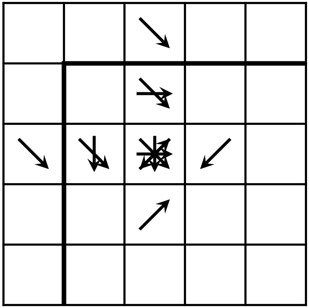
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.
6. Revert transport into halos: $\bar{f}_{\alpha,out}^{C,n} := \mathcal{T}^{-1}(\tilde{f}_{\alpha,out}^{C,n})$

Adaptive LBM

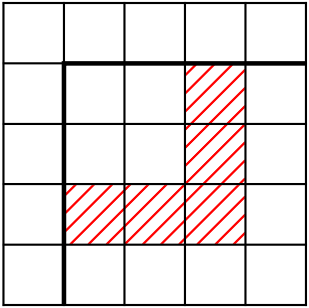
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.
6. Revert transport into halos: $\bar{f}_{\alpha,out}^{C,n} := \mathcal{T}^{-1}(\tilde{f}_{\alpha,out}^{C,n})$
7. Parallel synchronization of $f_\alpha^{C,n}, \bar{f}_{\alpha,out}^{C,n}$

Adaptive LBM

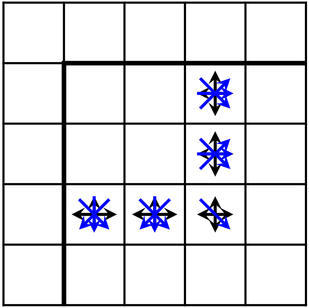
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.
6. Revert transport into halos: $\bar{f}_{\alpha,out}^{C,n} := \mathcal{T}^{-1}(\tilde{f}_{\alpha,out}^{C,n})$
7. Parallel synchronization of $f_\alpha^{C,n}, \bar{f}_{\alpha,out}^{C,n}$
8. Cell-wise update where correction is needed: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n}, \bar{f}_{\alpha,out}^{C,n})$

Adaptive LBM

1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2} := \mathcal{C}(\tilde{f}_\alpha^{f,n})$ in interior.
4. $\tilde{f}_\alpha^{f,n+1/2} := \mathcal{T}(f_\alpha^{f,n+1/2})$ on whole fine mesh. $f_\alpha^{f,n+1} := \mathcal{C}(\tilde{f}_\alpha^{f,n+1/2})$ in interior.



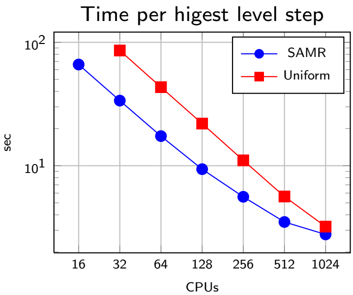
5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.
6. Revert transport into halos: $\bar{f}_{\alpha,out}^{C,n} := \mathcal{T}^{-1}(\tilde{f}_{\alpha,out}^{C,n})$
7. Parallel synchronization of $f_\alpha^{C,n}, \bar{f}_{\alpha,out}^{C,n}$
8. Cell-wise update where correction is needed: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n}, \bar{f}_{\alpha,out}^{C,n})$

Algorithm equivalent to [Chen et al., 2006].

AMROC strong scalability tests

3D wave propagation method with Roe scheme: spherical blast wave

- Tests run IBM BG/P (mode VN)



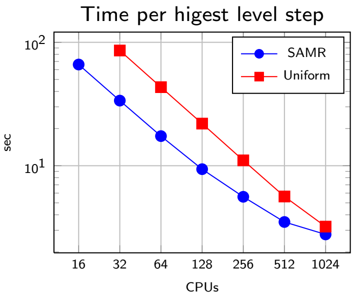
$64 \times 32 \times 32$ base grid, 2 additional levels with factors 2, 4; uniform $512 \times 256 \times 256 = 33.6 \cdot 10^6$ cells

Level	Grids	Cells
0	1709	65,536
1	1735	271,048
2	2210	7,190,208

AMROC strong scalability tests

3D wave propagation method with Roe scheme: spherical blast wave

- Tests run IBM BG/P (mode VN)

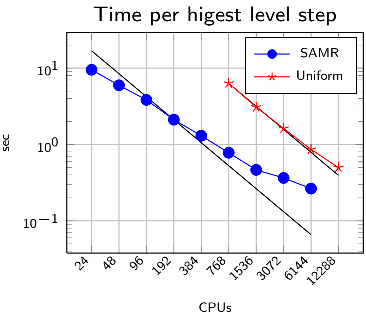


64 × 32 × 32 base grid, 2 additional levels with factors 2, 4; uniform $512 \times 256 \times 256 = 33.6 \cdot 10^6$ cells

Level	Grids	Cells
0	1709	65,536
1	1735	271,048
2	2210	7,190,208

3D SRT-lattice Boltzmann scheme: flow over rough surface of 19 × 13 × 2 spheres

- Tests run Cray XC30



360 × 240 × 108 base grid, 2 additional levels with factors 2, 4; uniform $1440 \times 1920 \times 432 = 1.19 \cdot 10^9$ cells

Level	Grids	Cells
0	788	9,331,200
1	21367	24,844,504
2	1728	10,838,016

Turbulence modeling

Pursue a large-eddy simulation approach with $\bar{f}_\alpha$ and $\bar{f}_\alpha^{eq}$, i.e.

1.) $\tilde{\bar{f}}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = \bar{f}_\alpha(\mathbf{x}, t)$

2.) $\bar{f}_\alpha(\cdot, t + \Delta t) = \tilde{\bar{f}}_\alpha(\cdot, t + \Delta t) + \frac{1}{\tau^\star} \Delta t \left(\tilde{\bar{f}}_\alpha^{eq}(\cdot, t + \Delta t) - \tilde{\bar{f}}_\alpha(\cdot, t + \Delta t) \right)$

Turbulence modeling

Pursue a large-eddy simulation approach with $\bar{f}_\alpha$ and $\bar{f}_\alpha^{eq}$, i.e.

1.) $\tilde{\bar{f}}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = \bar{f}_\alpha(\mathbf{x}, t)$

2.) $\bar{f}_\alpha(\cdot, t + \Delta t) = \tilde{\bar{f}}_\alpha(\cdot, t + \Delta t) + \frac{1}{\tau^\star} \Delta t \left(\tilde{\bar{f}}_\alpha^{eq}(\cdot, t + \Delta t) - \tilde{\bar{f}}_\alpha(\cdot, t + \Delta t) \right)$

Effective viscosity: $\nu^\star = \nu + \nu_t = \dfrac{1}{3}\left(\dfrac{\tau_L^\star}{\Delta t} - \dfrac{1}{2}\right) c\Delta x \quad \text{with} \quad \tau_L^\star = \tau_L + \tau_t$

Turbulence modeling

Pursue a large-eddy simulation approach with $\bar{f}_\alpha$ and $\bar{f}_\alpha^{eq}$, i.e.

1.) $\tilde{\bar{f}}_\alpha(\mathbf{x}+\mathbf{e}_\alpha\Delta t, t+\Delta t) = \bar{f}_\alpha(\mathbf{x}, t)$

2.) $\bar{f}_\alpha(\cdot, t+\Delta t) = \tilde{\bar{f}}_\alpha(\cdot, t+\Delta t) + \frac{1}{\tau^\star}\Delta t\left(\tilde{\bar{f}}_\alpha^{eq}(\cdot, t+\Delta t) - \tilde{\bar{f}}_\alpha(\cdot, t+\Delta t)\right)$

Effective viscosity: $\nu^\star = \nu + \nu_t = \dfrac{1}{3}\left(\dfrac{\tau_L^\star}{\Delta t} - \dfrac{1}{2}\right) c\Delta x \quad \text{with} \quad \tau_L^\star = \tau_L + \tau_t$

Use Smagorinsky model to evaluate ν_t, e.g., $\nu_t = (C_{sm}\Delta x)^2 \bar{S}$, where

$$\bar{S} = \sqrt{2\sum_{i,j} \bar{\mathbf{S}}_{ij}\bar{\mathbf{S}}_{ij}}$$

The filtered strain rate tensor $\bar{\mathbf{S}}_{ij} = (\partial_j \bar{u}_i + \partial_i \bar{u}_j)/2$ can be computed as a second moment as

$$\bar{\mathbf{S}}_{ij} = \frac{\Sigma_{ij}}{2\rho c_s^2 \tau_L^\star \left(1 - \frac{\omega_L \Delta t}{2}\right)} = \frac{1}{2\rho c_s^2 \tau_L^\star} \sum_\alpha \mathbf{e}_{\alpha i}\mathbf{e}_{\alpha j}(\bar{f}_\alpha^{eq} - \bar{f}_\alpha)$$

Turbulence modeling

Pursue a large-eddy simulation approach with $\bar{f}_\alpha$ and $\bar{f}_\alpha^{eq}$, i.e.

1.) $\tilde{\bar{f}}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = \bar{f}_\alpha(\mathbf{x}, t)$

2.) $\bar{f}_\alpha(\cdot, t + \Delta t) = \tilde{\bar{f}}_\alpha(\cdot, t + \Delta t) + \frac{1}{\tau^\star} \Delta t \left(\tilde{\bar{f}}_\alpha^{eq}(\cdot, t + \Delta t) - \tilde{\bar{f}}_\alpha(\cdot, t + \Delta t) \right)$

Effective viscosity: $\nu^\star = \nu + \nu_t = \dfrac{1}{3} \left(\dfrac{\tau_L^\star}{\Delta t} - \dfrac{1}{2} \right) c \Delta x$ with $\tau_L^\star = \tau_L + \tau_t$

Use Smagorinsky model to evaluate ν_t, e.g., $\nu_t = (C_{sm} \Delta x)^2 \bar{S}$, where

$$\bar{S} = \sqrt{2 \sum_{i,j} \bar{\mathbf{S}}_{ij} \bar{\mathbf{S}}_{ij}}$$

The filtered strain rate tensor $\bar{\mathbf{S}}_{ij} = (\partial_j \bar{u}_i + \partial_i \bar{u}_j)/2$ can be computed as a second moment as

$$\bar{\mathbf{S}}_{ij} = \frac{\Sigma_{ij}}{2 \rho c_s^2 \tau_L^\star \left(1 - \frac{\omega_L \Delta t}{2} \right)} = \frac{1}{2 \rho c_s^2 \tau_L^\star} \sum_\alpha \mathbf{e}_{\alpha i} \mathbf{e}_{\alpha j} (\bar{f}_\alpha^{eq} - \bar{f}_\alpha)$$

τ_t can be obtained as [Yu, 2004, Hou et al., 1996]

$$\tau_t = \frac{1}{2} \left(\sqrt{\tau_L^2 + 18 \sqrt{2} (\rho_0 c^2)^{-1} C_{sm}^2 \Delta x \bar{S}} - \tau_L \right)$$

Homogeneous isotropic turbulence

- Fourier representation
- Periodic boundaries, uniform mesh
- Use of external forcing term, i.e., result independent of initial conditions

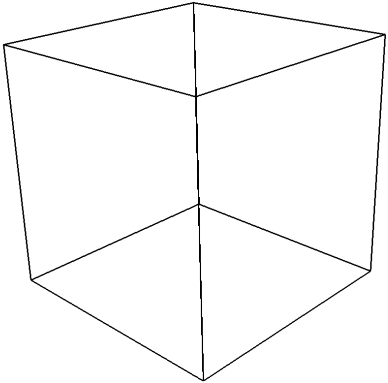
Forcing:

$$F_x = 2A\left(\frac{\kappa_y \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_y = -A\left(\frac{\kappa_x \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_z = -A\left(\frac{\kappa_x \kappa_y}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

Iso-surface $||\mathbf{u}||/\langle u_{rms}\rangle = 2$



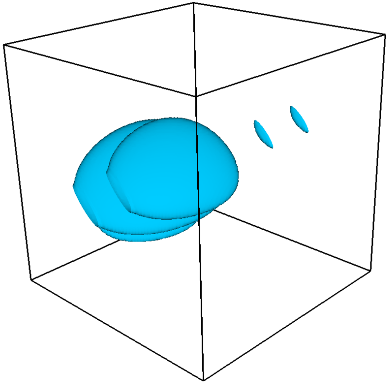
with phase

$G(\kappa_x, \kappa_y, \kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right)$ for $(0 < \kappa_i \leq 2)$ and ϕ being a random phase value.

Homogeneous isotropic turbulence

- Fourier representation
- Periodic boundaries, uniform mesh
- Use of external forcing term, i.e., result independent of initial conditions

Iso-surface $||\mathbf{u}||/\langle u_{rms}\rangle = 2$



Forcing:

$$F_x = 2A\left(\frac{\kappa_y\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

$$F_y = -A\left(\frac{\kappa_x\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

$$F_z = -A\left(\frac{\kappa_x\kappa_y}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

with phase

$G(\kappa_x,\kappa_y,\kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right)$ for $(0 < \kappa_i \leq 2)$ and ϕ being a random phase value.

Homogeneous isotropic turbulence

- Fourier representation
- Periodic boundaries, uniform mesh
- Use of external forcing term, i.e., result independent of initial conditions

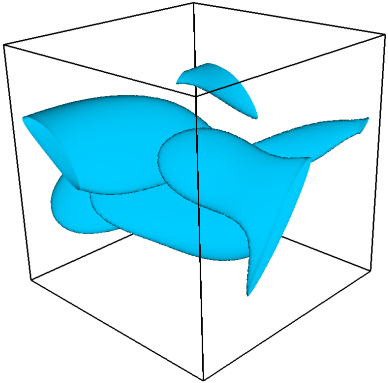
Forcing:

$$F_x = 2A\left(\frac{\kappa_y \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_y = -A\left(\frac{\kappa_x \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_z = -A\left(\frac{\kappa_x \kappa_y}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

Iso-surface $||\mathbf{u}||/\langle u_{rms}\rangle = 2$



with phase

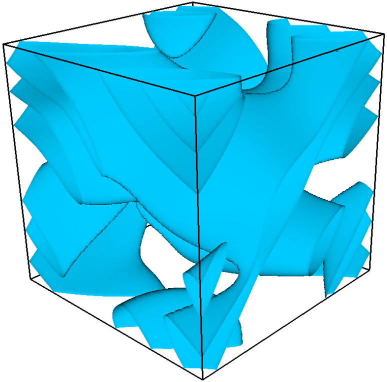
$$G(\kappa_x, \kappa_y, \kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right) \text{ for } (0 < \kappa_i \leq 2) \text{ and } \phi$$

being a random phase value.

Homogeneous isotropic turbulence

- Fourier representation
- Periodic boundaries, uniform mesh
- Use of external forcing term, i.e., result independent of initial conditions

Iso-surface $||\mathbf{u}||/\langle u_{rms}\rangle = 2$



Forcing:

$$F_x = 2A\left(\frac{\kappa_y\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

$$F_y = -A\left(\frac{\kappa_x\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

$$F_z = -A\left(\frac{\kappa_x\kappa_y}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

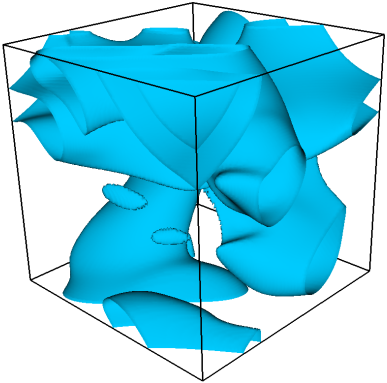
with phase

$G(\kappa_x,\kappa_y,\kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right)$ for $(0 < \kappa_i \leq 2)$ and ϕ being a random phase value.

Homogeneous isotropic turbulence

- Fourier representation
- Periodic boundaries, uniform mesh
- Use of external forcing term, i.e., result independent of initial conditions

Iso-surface $||\mathbf{u}||/\langle u_{rms}\rangle = 2$



Forcing:

$$F_x = 2A\left(\frac{\kappa_y \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_y = -A\left(\frac{\kappa_x \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_z = -A\left(\frac{\kappa_x \kappa_y}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

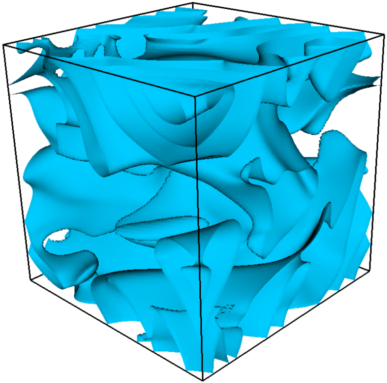
with phase

$G(\kappa_x, \kappa_y, \kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right)$ for $(0 < \kappa_i \leq 2)$ and ϕ being a random phase value.

Homogeneous isotropic turbulence

- Fourier representation
- Periodic boundaries, uniform mesh
- Use of external forcing term, i.e., result independent of initial conditions

Iso-surface $||\mathbf{u}||/\langle u_{rms}\rangle = 2$



Forcing:

$$F_x = 2A\left(\frac{\kappa_y\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

$$F_y = -A\left(\frac{\kappa_x\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

$$F_z = -A\left(\frac{\kappa_x\kappa_y}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

with phase

$G(\kappa_x,\kappa_y,\kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right)$ for $(0 < \kappa_i \leq 2)$ and ϕ being a random phase value.

Homogeneous isotropic turbulence

- Fourier representation
- Periodic boundaries, uniform mesh
- Use of external forcing term, i.e., result independent of initial conditions

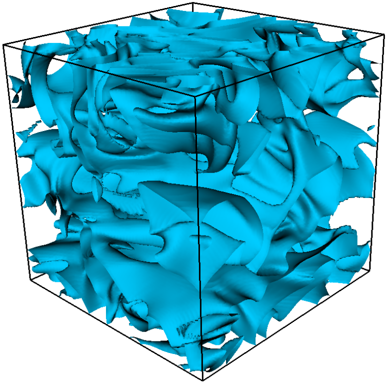
Forcing:

$$F_x = 2A\left(\frac{\kappa_y \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_y = -A\left(\frac{\kappa_x \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_z = -A\left(\frac{\kappa_x \kappa_y}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

Iso-surface $||\mathbf{u}||/\langle u_{rms}\rangle = 2$



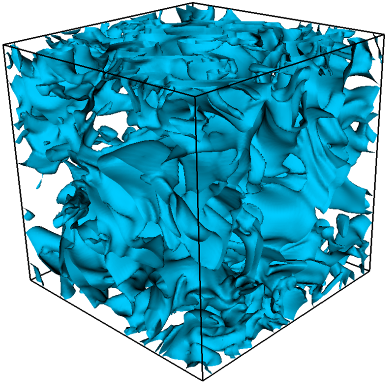
with phase

$G(\kappa_x, \kappa_y, \kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right)$ for $(0 < \kappa_i \leq 2)$ and ϕ being a random phase value.

Homogeneous isotropic turbulence

- Fourier representation
- Periodic boundaries, uniform mesh
- Use of external forcing term, i.e., result independent of initial conditions

Iso-surface $||\mathbf{u}||/\langle u_{rms}\rangle = 2$



Forcing:

$$F_x = 2A\left(\frac{\kappa_y\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

$$F_y = -A\left(\frac{\kappa_x\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

$$F_z = -A\left(\frac{\kappa_x\kappa_y}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

with phase

$G(\kappa_x,\kappa_y,\kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right)$ for $(0 < \kappa_i \leq 2)$ and ϕ being a random phase value.

Homogeneous isotropic turbulence

- Fourier representation
- Periodic boundaries, uniform mesh
- Use of external forcing term, i.e., result independent of initial conditions

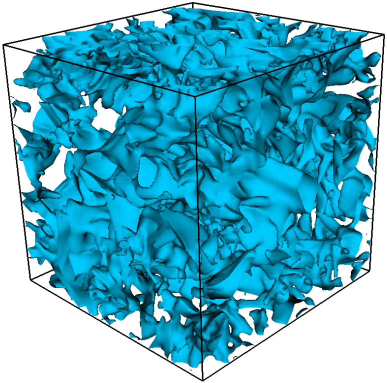
Forcing:

$$F_x = 2A\left(\frac{\kappa_y \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_y = -A\left(\frac{\kappa_x \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_z = -A\left(\frac{\kappa_x \kappa_y}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

Iso-surface $||\mathbf{u}||/\langle u_{rms}\rangle = 2$



with phase

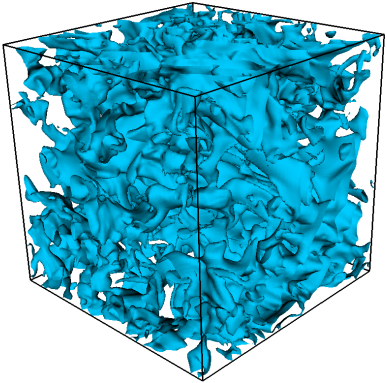
$$G(\kappa_x, \kappa_y, \kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right) \text{ for } (0 < \kappa_i \le 2) \text{ and } \phi$$

being a random phase value.

Homogeneous isotropic turbulence

- Fourier representation
- Periodic boundaries, uniform mesh
- Use of external forcing term, i.e., result independent of initial conditions

Iso-surface $||\mathbf{u}||/\langle u_{rms}\rangle = 2$



Forcing:

$$F_x = 2A\left(\frac{\kappa_y \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

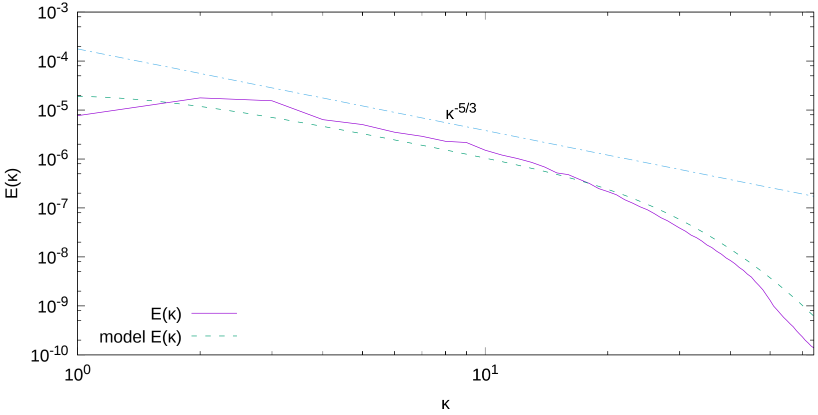
$$F_y = -A\left(\frac{\kappa_x \kappa_z}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

$$F_z = -A\left(\frac{\kappa_x \kappa_y}{|\kappa|^2}\right) G(\kappa_x, \kappa_y, \kappa_z)$$

with phase

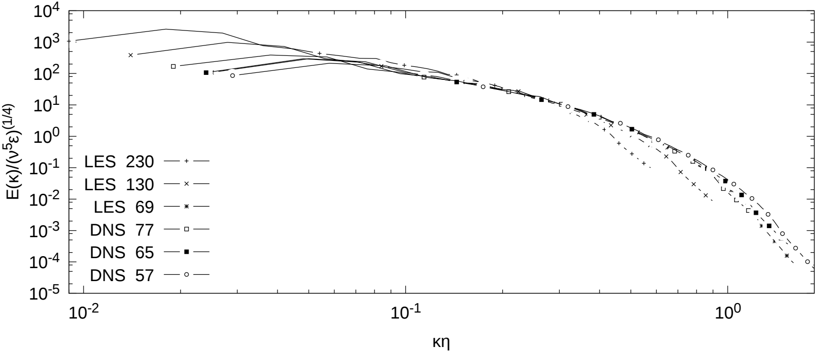
$G(\kappa_x, \kappa_y, \kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right)$ for $(0 < \kappa_i \leq 2)$ and ϕ being a random phase value.

Results



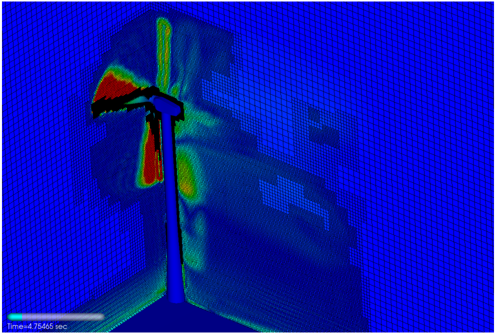
Time-averaged energy spectrum (solid line) [$N = 128^3$ cells, $\nu = 3e^{-5}\ \mathrm{m}^2/\mathrm{s}$] against a modelled one (dashed line and the -5/3 power law (dot-dashed line).

Results - Kolmogorov spectrum function



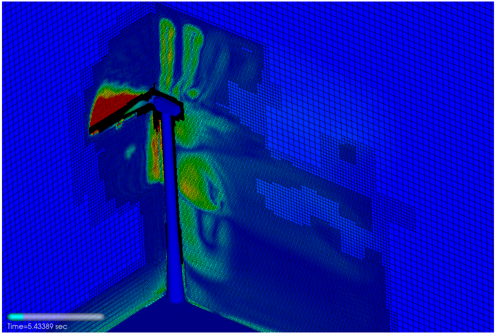
Time-averaged Kolmogorov spectra of DNS and LES for different Re numbers

Single Vestas V27



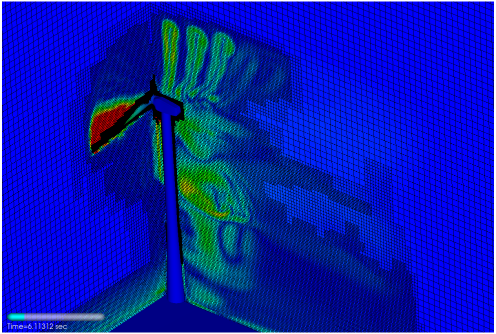
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



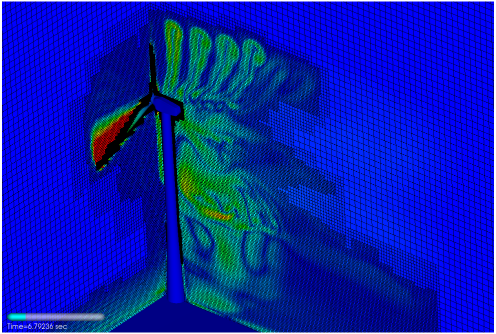
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



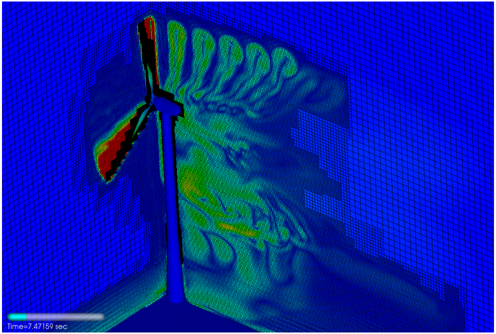
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



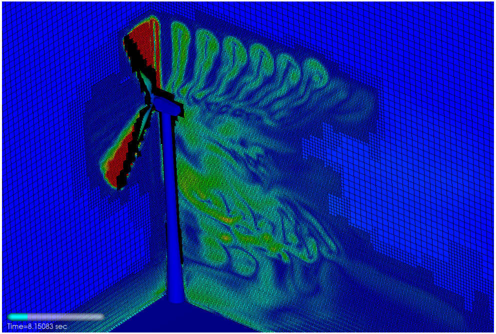
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



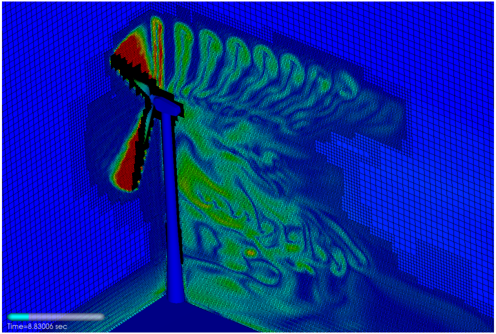
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



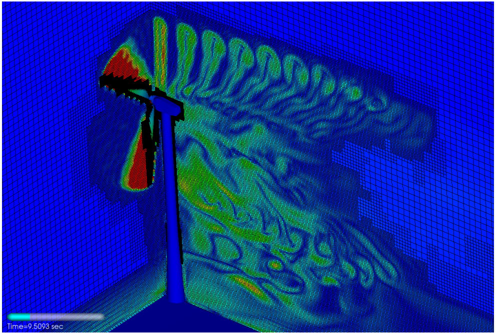
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



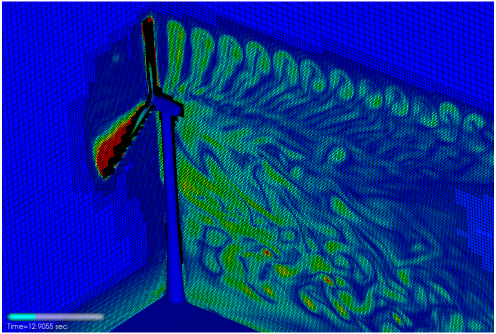
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



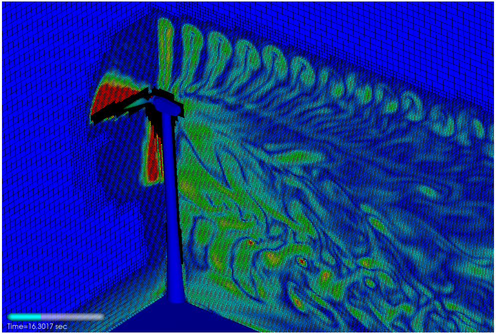
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



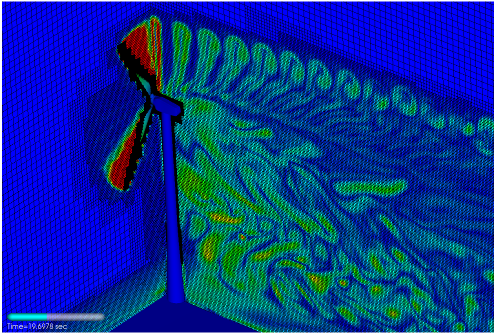
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



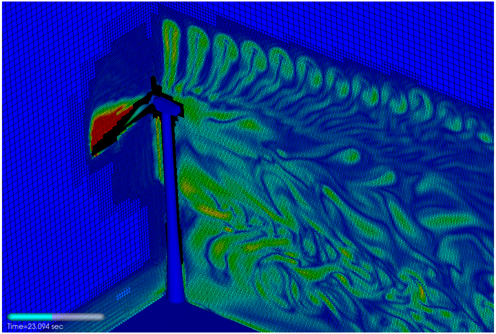
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



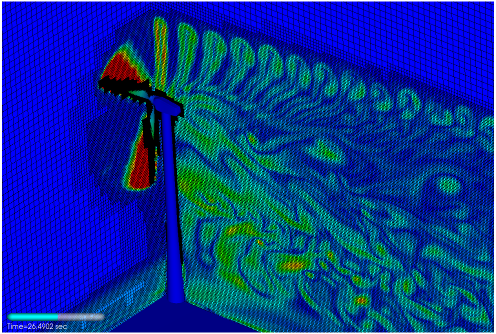
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



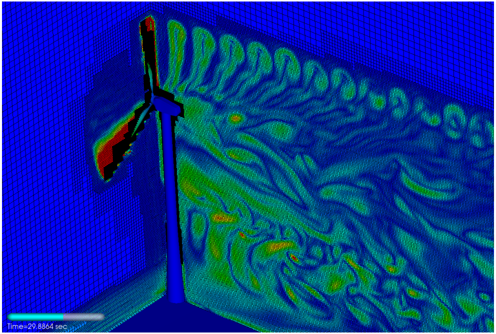
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



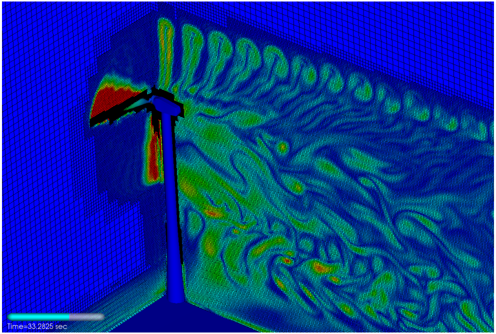
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

Single Vestas V27



- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

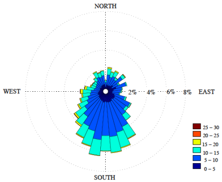
Single Vestas V27

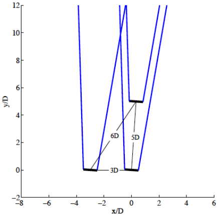


- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$, TSR=5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
  item ~ 24 time steps for 1^o rotation
- Validation results: Mexico rotor [Deiterding and Wood, 2016b], [Deiterding and Wood, 2016a]

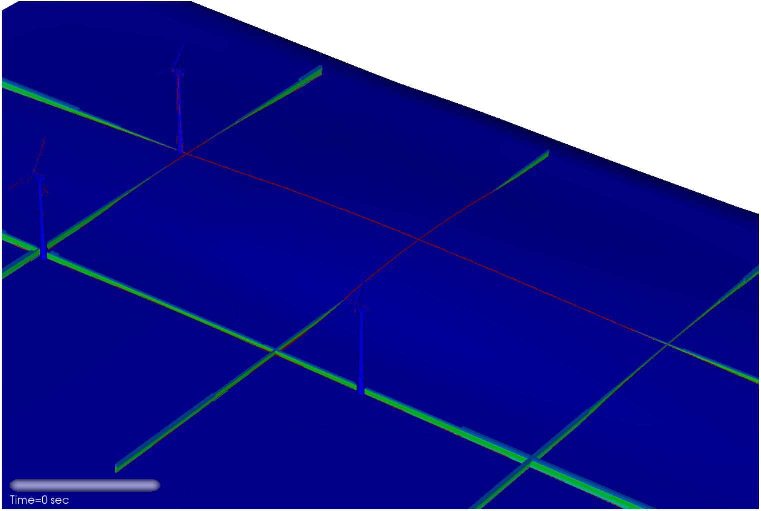
Simulation of the SWIFT array

- Three Vestas V27 turbines (geometric details prototypical). 225 kW power generation at wind speeds 14 to 25 $\mathrm{m/s}$ (then cut-off)
- Prescribed motion of rotor with 33 and 43 rpm. Inflow velocity 8 and 25 $\mathrm{m/s}$
- TSR: 5.84 and 2.43, $\mathrm{Re}_r \approx 919,700$ and 1,208,000
- Simulation domain $448\,\mathrm{m} \times 240\,\mathrm{m} \times 100\,\mathrm{m}$
- Base mesh $448 \times 240 \times 100$ cells with refinement factors 2, 2,4. Resolution of rotor and tower $\Delta x = 6.25\,\mathrm{cm}$
- 94,224 highest level iterations to $t_e = 40\,\mathrm{s}$ computed, then statistics are gathered for $10\,\mathrm{s}$ [Deiterding and Wood, 2016a]



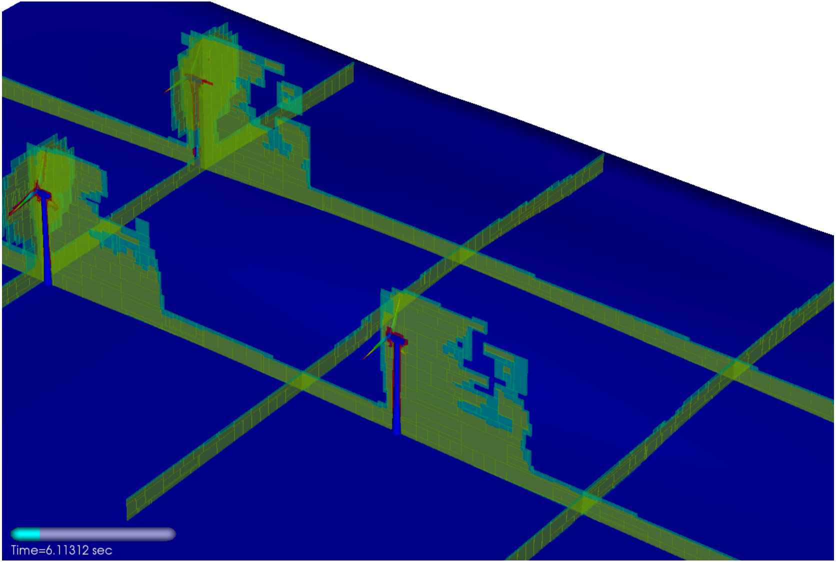


Vorticity development – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$

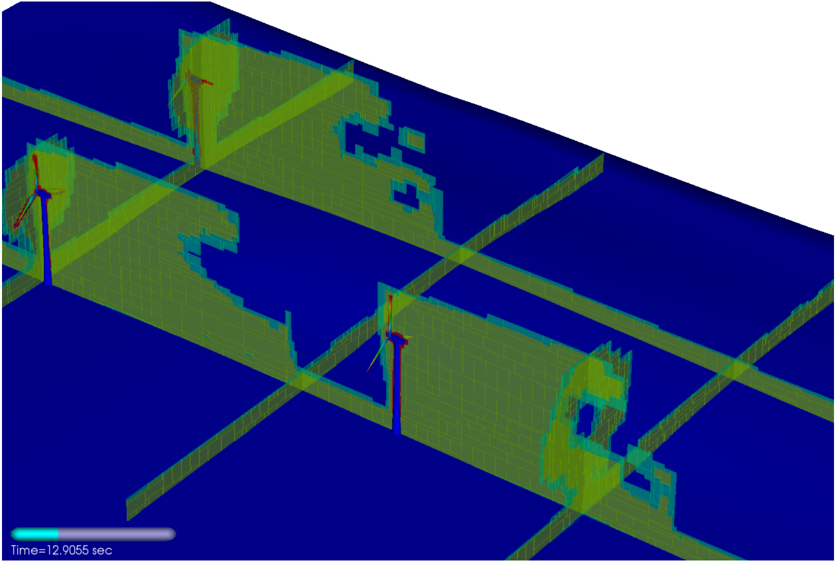


- Refinement of wake up to level 2 ($\Delta x = 25\,\mathrm{cm}$).
- Vortex break-up before 2nd turbine is reached.

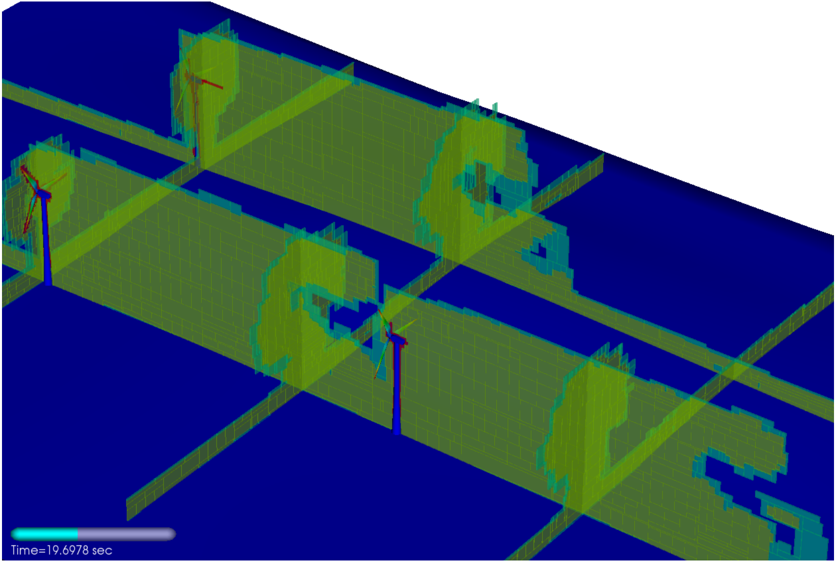
Refinement – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



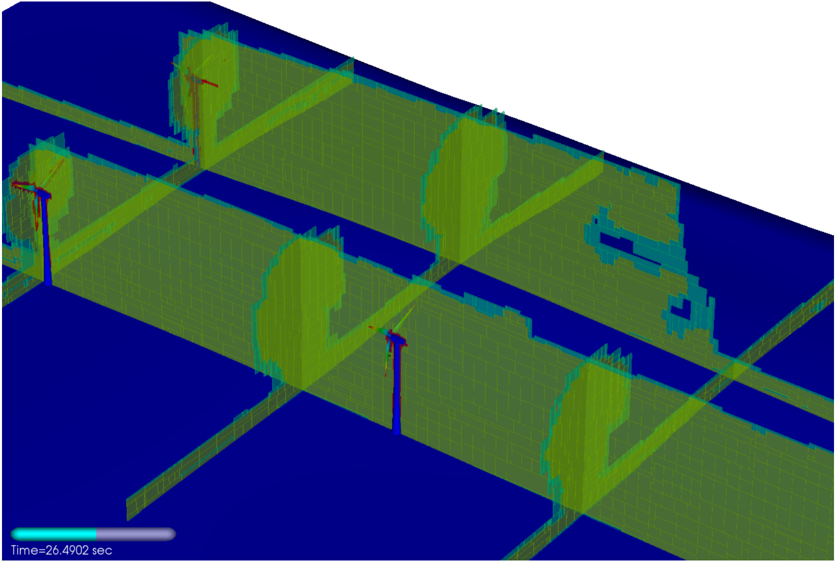
Refinement – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



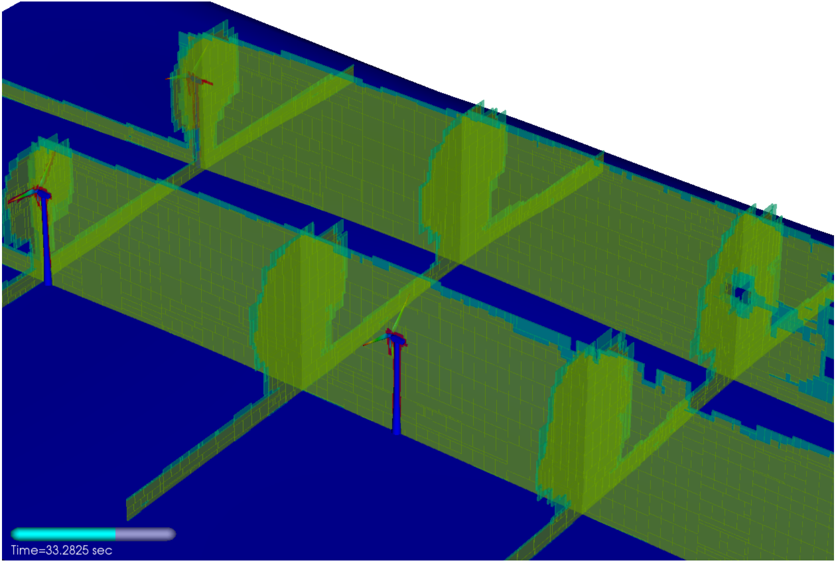
Refinement – inflow at 0°, $u = 8\,\mathrm{m/s}$, 33 rpm



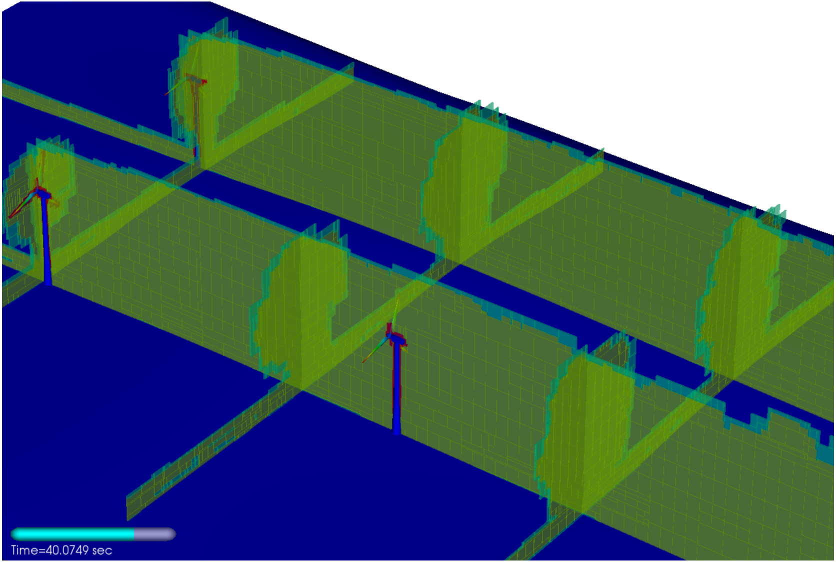
Refinement – inflow at 0°, $u = 8\,\mathrm{m/s}$, 33 rpm



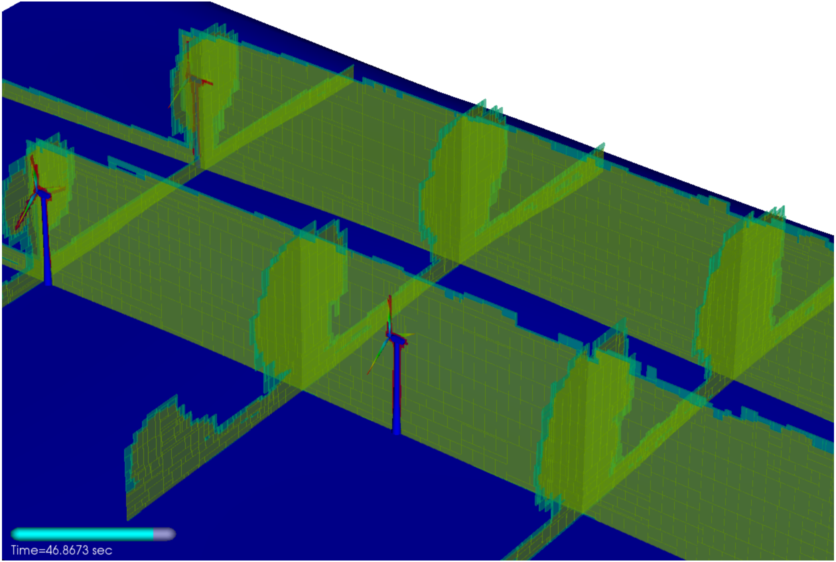
Refinement – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



Refinement – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$

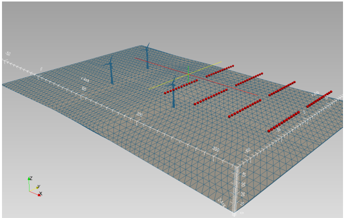


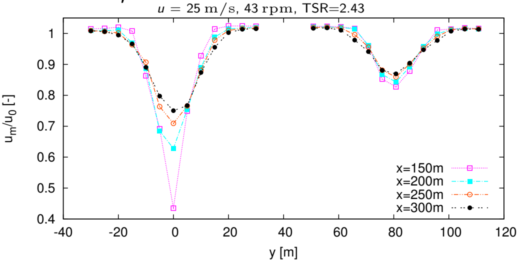
Refinement – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$

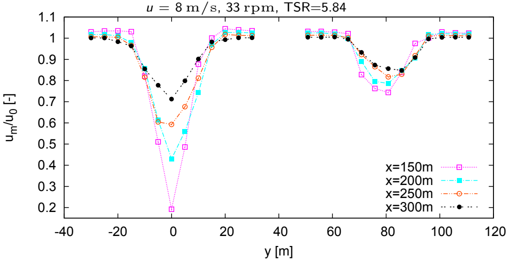


Mean point values – inflow at 0^o,

- Turbines located at $(0,0,0)$, $(135,0,0)$, $(-5.65, 80.80, 0)$
- Lines of 13 sensors with $\Delta y = 5\,\mathrm{m}$, $z = 37\,\mathrm{m}$ (approx. center of rotor)
- u and p measured over $[40\,\mathrm{s}, 50\,\mathrm{s}]$ (1472 level-0 time steps) and averaged



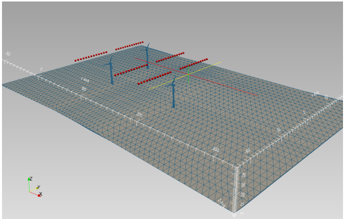


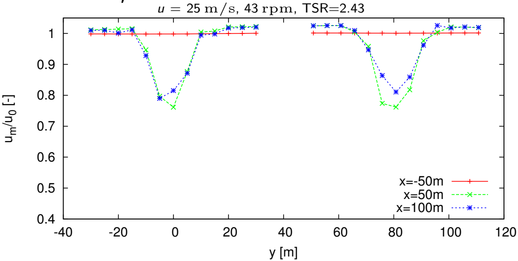


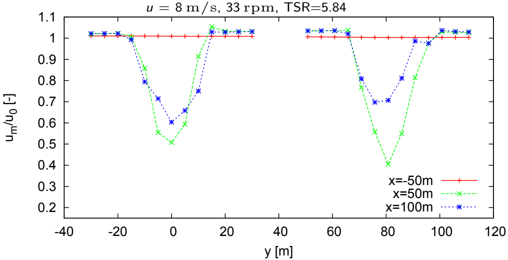
- Velocity deficits larger for higher TSR.

Mean point values – inflow at 0^o,

- Turbines located at $(0, 0, 0)$, $(135, 0, 0)$, $(-5.65, 80.80, 0)$
- Lines of 13 sensors with $\Delta y = 5\,\mathrm{m}$, $z = 37\,\mathrm{m}$ (approx. center of rotor)
- u and p measured over $[40\,\mathrm{s}, 50\,\mathrm{s}]$ (1472 level-0 time steps) and averaged

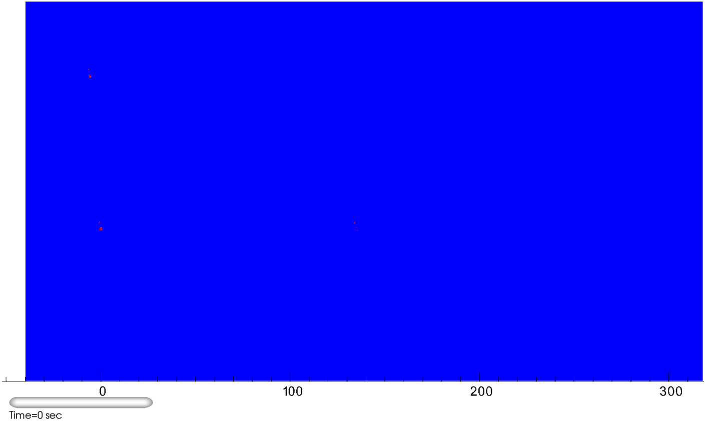






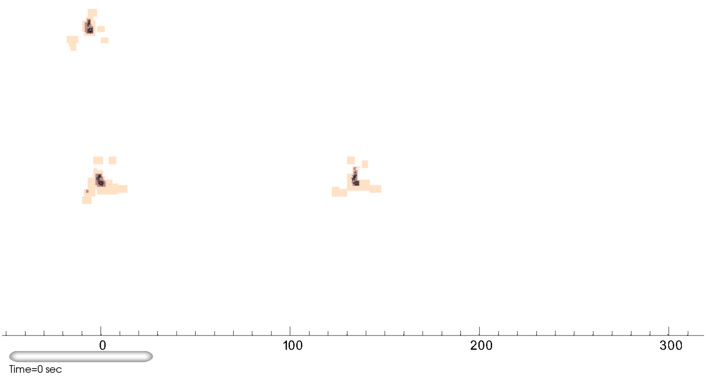
- Velocity deficits larger for higher TSR.
- Velocity deficit before 2nd turbine more homogenous for small TSR.

Vorticity – inflow at 30^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



- Top view in plane in z-direction at $30\,\mathrm{m}$ (hub height)
- Turbine hub and inflow at 30^o yaw leads to off-axis wake impact.
- 160 cores Intel-Xeon E5 2.6 GHz, $33.03\,\mathrm{h}$ wall time for interval $[50, 60]\,\mathrm{s}$ (including gathering of statistical data)

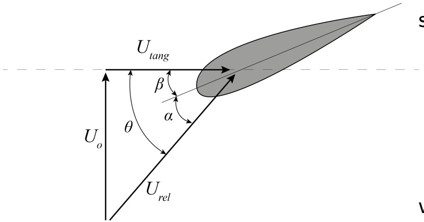
Levels – inflow at 30°, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



- At $63.8\,\mathrm{s}$ approximately 167M cells used vs. 44 billion (factor 264)
- $\sim 6.01\,\mathrm{h}$ per revolution ($961\,\mathrm{h}$ CPU) $\longrightarrow \sim 5.74\,\mathrm{h}$ CPU/1M cells/revolution

Level	Grids	Cells
0	2,463	10,752,000
1	6,464	20,674,760
2	39,473	131,018,832
3	827	4,909,632

Blade element modeling



Transformation into blade coordinate system:

$$U_{tang} = wr(1 + a')$$

$$U_o = U_\infty(1 - a)$$

$$U_{rel} = \sqrt{w^2 r^2 (1 + a')^2 + U_\infty^2 (1 - a)^2}$$

where a is the axial and a' the tangential induction factor

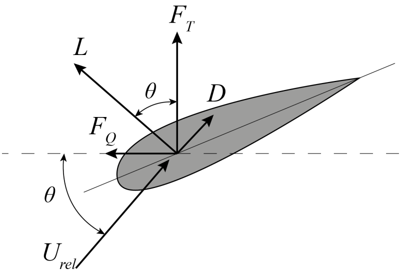
Local aerodynamic forces on a 2D blade profile:

$$L = \frac{1}{2}\rho U_{rel}^2 c C_l \partial r$$

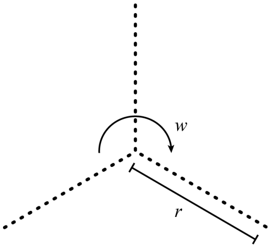
$$D = \frac{1}{2}\rho U_{rel}^2 c C_d \partial r$$

Axial and radial force:

$$F_T = L\cos\theta + D\sin\theta, \quad F_Q = L\sin\theta - D\cos\theta$$



Actuator line model

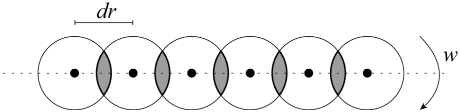


Gaussian spreading function [Sørensen et al., 1998]

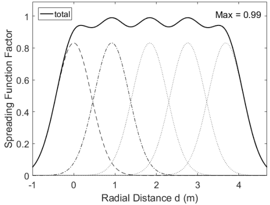
$$f(d) = \frac{1}{\varepsilon^3 \pi^{\frac{3}{2}}} \exp\left(-\frac{d}{\varepsilon}\right)^2$$

Distance d between cell midpoint and ith actuator point

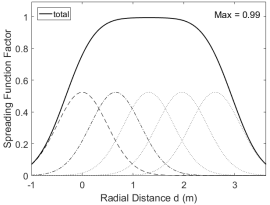
Overlapping actuator points



Appropriate choice of ε and dr is essential:

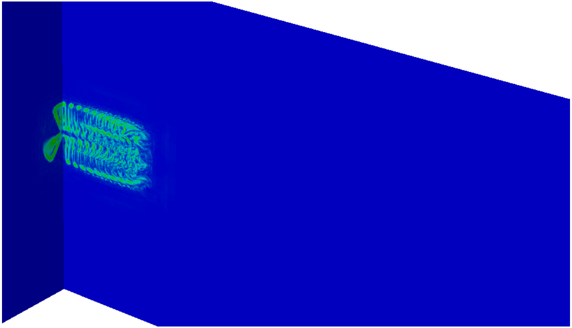


$\varepsilon = 0.6,\ dr = 0.92\,\mathrm{m}$



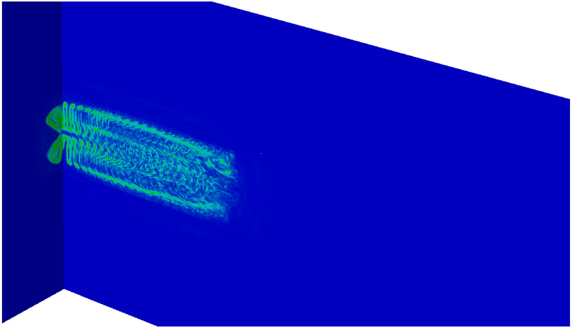
$\varepsilon = 0.7,\ dr = 0.65\,\mathrm{m}$

Simulation of single V27 rotor



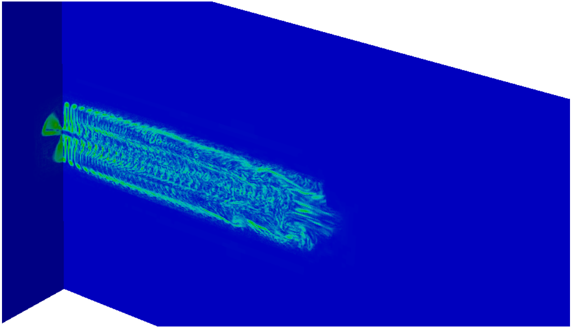
- $u_\infty = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$, TSR: 5.84
- Simulation domain $320\,\mathrm{m} \times 160\,\mathrm{m} \times 160\,\mathrm{m}$
- Base mesh $80 \times 40 \times 40$ cells with refinement factors 2, 2, 4. Finest resolution of rotor and tower $\Delta x = 25\,\mathrm{cm}$ (same as before for wake)
- $t_e = 50\,\mathrm{s}$ computed. $96\,\mathrm{h}$ CPU on 12 cores Intel-Xeon-E5-2.10GHz

Simulation of single V27 rotor



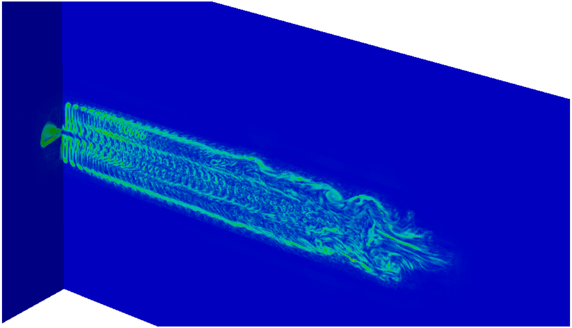
- $u_\infty = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$, TSR: 5.84
- Simulation domain $320\,\mathrm{m} \times 160\,\mathrm{m} \times 160\,\mathrm{m}$
- Base mesh $80 \times 40 \times 40$ cells with refinement factors 2, 2, 4. Finest resolution of rotor and tower $\Delta x = 25\,\mathrm{cm}$ (same as before for wake)
- $t_e = 50\,\mathrm{s}$ computed. $96\,\mathrm{h}$ CPU on 12 cores Intel-Xeon-E5-2.10GHz

Simulation of single V27 rotor



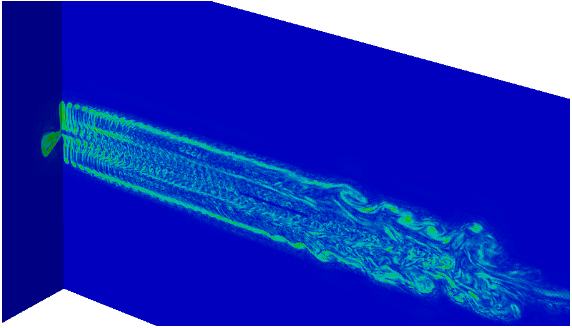
- $u_\infty = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$, TSR: 5.84
- Simulation domain $320\,\mathrm{m} \times 160\,\mathrm{m} \times 160\,\mathrm{m}$
- Base mesh $80 \times 40 \times 40$ cells with refinement factors 2, 2, 4. Finest resolution of rotor and tower $\Delta x = 25\,\mathrm{cm}$ (same as before for wake)
- $t_e = 50\,\mathrm{s}$ computed. $96\,\mathrm{h}$ CPU on 12 cores Intel-Xeon-E5-2.10GHz

Simulation of single V27 rotor



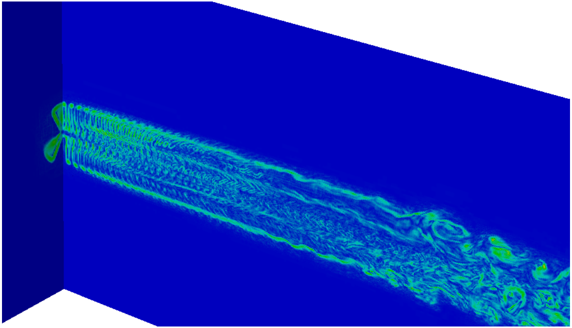
- $u_\infty = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$, TSR: 5.84
- Simulation domain $320\,\mathrm{m} \times 160\,\mathrm{m} \times 160\,\mathrm{m}$
- Base mesh $80 \times 40 \times 40$ cells with refinement factors 2, 2, 4. Finest resolution of rotor and tower $\Delta x = 25\,\mathrm{cm}$ (same as before for wake)
- $t_e = 50\,\mathrm{s}$ computed. $96\,\mathrm{h}$ CPU on 12 cores Intel-Xeon-E5-2.10GHz

Simulation of single V27 rotor



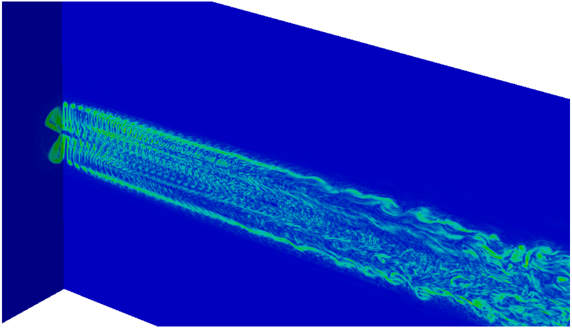
- $u_\infty = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$, TSR: 5.84
- Simulation domain $320\,\mathrm{m} \times 160\,\mathrm{m} \times 160\,\mathrm{m}$
- Base mesh $80 \times 40 \times 40$ cells with refinement factors 2, 2, 4. Finest resolution of rotor and tower $\Delta x = 25\,\mathrm{cm}$ (same as before for wake)
- $t_e = 50\,\mathrm{s}$ computed. $96\,\mathrm{h}$ CPU on 12 cores Intel-Xeon-E5-2.10GHz

Simulation of single V27 rotor



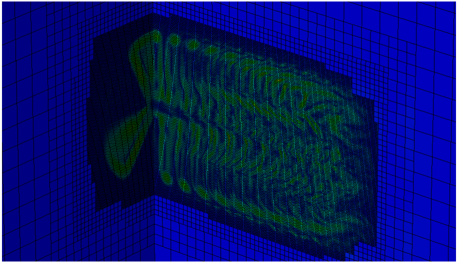
- $u_\infty = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$, TSR: 5.84
- Simulation domain $320\,\mathrm{m} \times 160\,\mathrm{m} \times 160\,\mathrm{m}$
- Base mesh $80 \times 40 \times 40$ cells with refinement factors 2, 2, 4. Finest resolution of rotor and tower $\Delta x = 25\,\mathrm{cm}$ (same as before for wake)
- $t_e = 50\,\mathrm{s}$ computed. $96\,\mathrm{h}$ CPU on 12 cores Intel-Xeon-E5-2.10GHz

Simulation of single V27 rotor



- $u_\infty = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$, TSR: 5.84
- Simulation domain $320\,\mathrm{m} \times 160\,\mathrm{m} \times 160\,\mathrm{m}$
- Base mesh $80 \times 40 \times 40$ cells with refinement factors 2, 2, 4. Finest resolution of rotor and tower $\Delta x = 25\,\mathrm{cm}$ (same as before for wake)
- $t_e = 50\,\mathrm{s}$ computed. $96\,\mathrm{h}$ CPU on 12 cores Intel-Xeon-E5-2.10GHz

Simulation of single V27 rotor - II



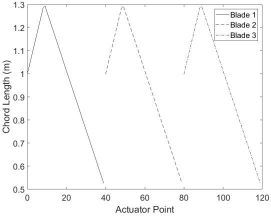
Actuator modelling:

- 3 actuator lines with 40 points. Inner radius $0.5\,\mathrm{m}$, outer radius $13.5\,\mathrm{m}$, $\varepsilon = 2\,\mathrm{m}$, $dr = 0.325\,\mathrm{m}$
- Tip loss correction by [Shen et al., 2005]

$$F_1 = \frac{2}{\pi}\cos^{-1}\left[\exp\left(-g\frac{B(R-r)}{2r\sin\phi}\right)\right]$$

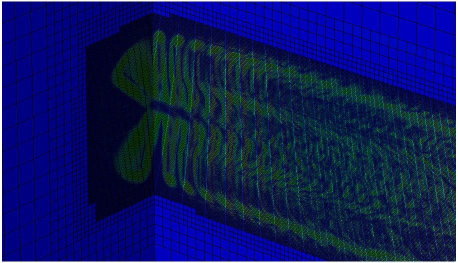
with $g = \exp[0.125(B\lambda - 21)] + 0.1$

- Chord length modeled roughly along actual blade



Chord distribution over 3 blades

Simulation of single V27 rotor - II



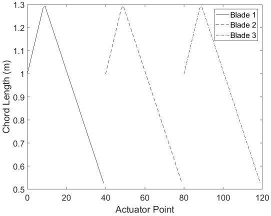
Actuator modelling:

- 3 actuator lines with 40 points. Inner radius $0.5\,\mathrm{m}$, outer radius $13.5\,\mathrm{m}$, $\varepsilon = 2\,\mathrm{m}$, $dr = 0.325\,\mathrm{m}$
- Tip loss correction by [Shen et al., 2005]

$$F_1 = \frac{2}{\pi}\cos^{-1}\left[\exp\left(-g\frac{B(R-r)}{2r\sin\phi}\right)\right]$$

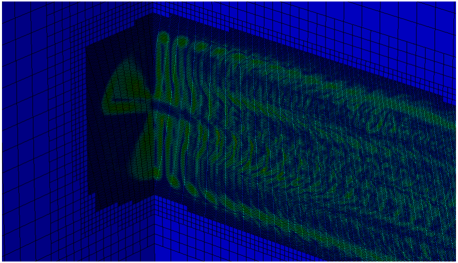
with $g = \exp[0.125(B\lambda - 21)] + 0.1$

- Chord length modeled roughly along actual blade



Chord distribution over 3 blades

Simulation of single V27 rotor - II



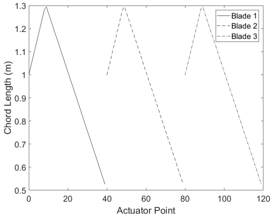
Actuator modelling:

- 3 actuator lines with 40 points. Inner radius $0.5\,\mathrm{m}$, outer radius $13.5\,\mathrm{m}$, $\varepsilon = 2\,\mathrm{m}$, $dr = 0.325\,\mathrm{m}$
- Tip loss correction by [Shen et al., 2005]

$$F_1 = \frac{2}{\pi}\cos^{-1}\left[\exp\left(-g\frac{B(R-r)}{2r\sin\phi}\right)\right]$$

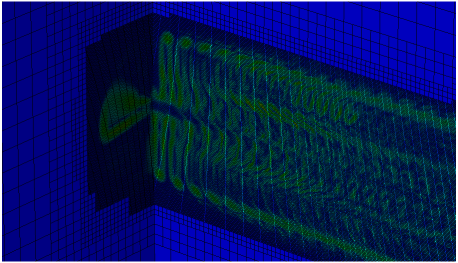
with $g = \exp[0.125(B\lambda - 21)] + 0.1$

- Chord length modeled roughly along actual blade



Chord distribution over 3 blades

Simulation of single V27 rotor - II



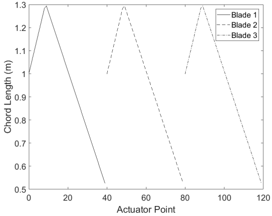
Actuator modelling:

- 3 actuator lines with 40 points. Inner radius $0.5\,\mathrm{m}$, outer radius $13.5\,\mathrm{m}$, $\varepsilon = 2\,\mathrm{m}$, $dr = 0.325\,\mathrm{m}$
- Tip loss correction by [Shen et al., 2005]

$$F_1 = \frac{2}{\pi}\cos^{-1}\left[\exp\left(-g\frac{B(R-r)}{2r\sin\phi}\right)\right]$$

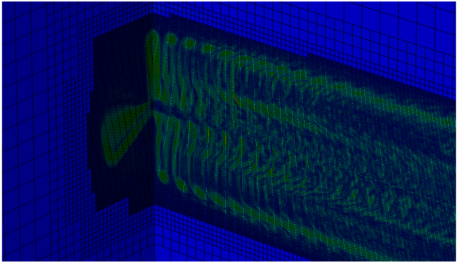
with $g = \exp[0.125(B\lambda - 21)] + 0.1$

- Chord length modeled roughly along actual blade



Chord distribution over 3 blades

Simulation of single V27 rotor - II



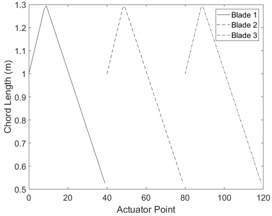
Actuator modelling:

- 3 actuator lines with 40 points. Inner radius $0.5\,\mathrm{m}$, outer radius $13.5\,\mathrm{m}$, $\varepsilon = 2\,\mathrm{m}$, $dr = 0.325\,\mathrm{m}$
- Tip loss correction by [Shen et al., 2005]

$$F_1 = \frac{2}{\pi}\cos^{-1}\left[\exp\left(-g\frac{B(R-r)}{2r\sin\phi}\right)\right]$$

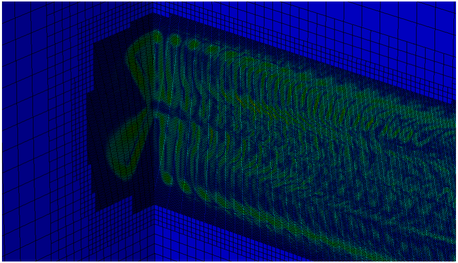
with $g = \exp[0.125(B\lambda - 21)] + 0.1$

- Chord length modeled roughly along actual blade



Chord distribution over 3 blades

Simulation of single V27 rotor - II



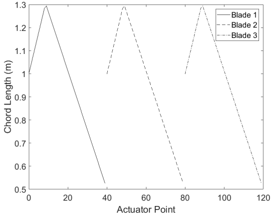
Actuator modelling:

- 3 actuator lines with 40 points. Inner radius $0.5\,\mathrm{m}$, outer radius $13.5\,\mathrm{m}$, $\varepsilon = 2\,\mathrm{m}$, $dr = 0.325\,\mathrm{m}$
- Tip loss correction by [Shen et al., 2005]

$$F_1 = \frac{2}{\pi}\cos^{-1}\left[\exp\left(-g\frac{B(R-r)}{2r\sin\phi}\right)\right]$$

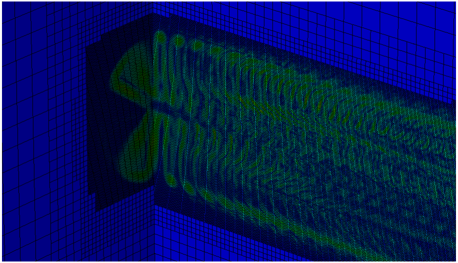
with $g = \exp[0.125(B\lambda - 21)] + 0.1$

- Chord length modeled roughly along actual blade



Chord distribution over 3 blades

Simulation of single V27 rotor - II



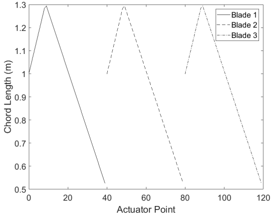
Actuator modelling:

- 3 actuator lines with 40 points. Inner radius $0.5\,\mathrm{m}$, outer radius $13.5\,\mathrm{m}$, $\varepsilon = 2\,\mathrm{m}$, $dr = 0.325\,\mathrm{m}$
- Tip loss correction by [Shen et al., 2005]

$$F_1 = \frac{2}{\pi}\cos^{-1}\left[\exp\left(-g\frac{B(R-r)}{2r\sin\phi}\right)\right]$$

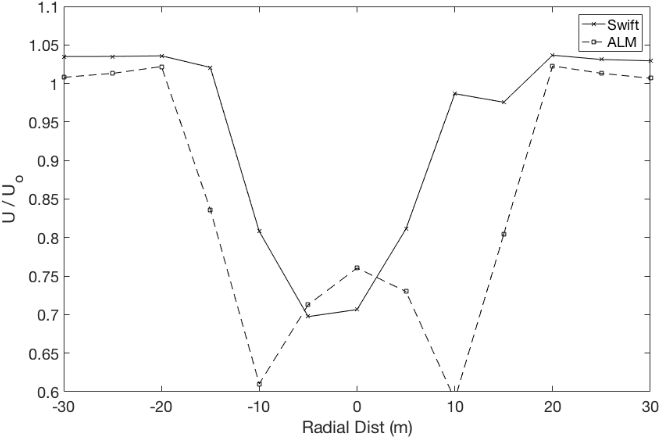
with $g = \exp[0.125(B\lambda - 21)] + 0.1$

- Chord length modeled roughly along actual blade



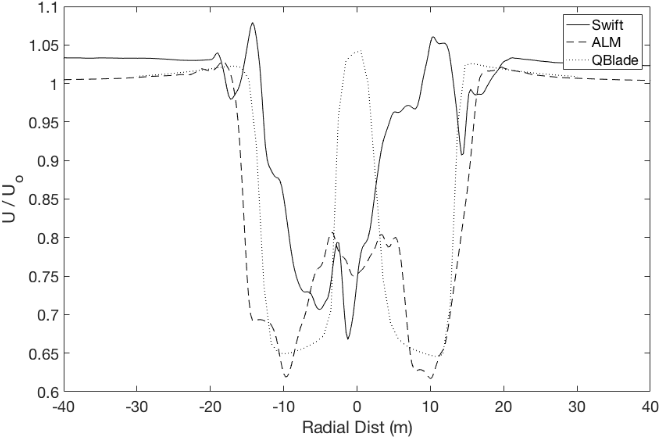
Chord distribution over 3 blades

Axial velocity comparison



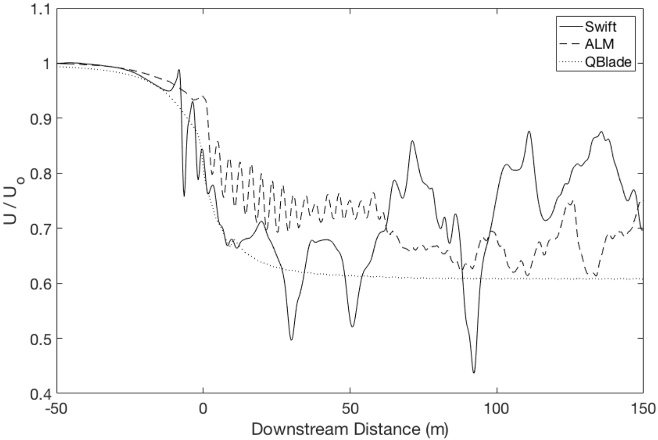
Mean axial velocity in the wake 100m downstream

Axial velocity profiles at $t = 43\,\mathrm{s}$



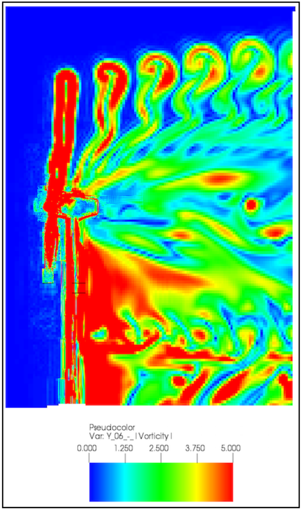
At hub height 100m downstream

Axial velocity profiles at $t = 43\,\mathrm{s}$

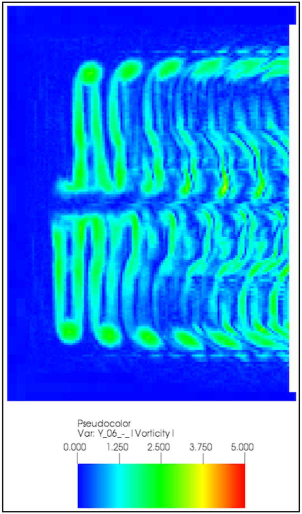


Downstream at rotor center

Vorticity between -5 and 25m downstream, $t = 43\,\mathrm{s}$



SWIFT



ALM

Conclusions

- Developed a first verified version of a block-based dynamically adaptive LBM for wind turbine wake aerodynamics using actuator models (disc model not shown here)
- Actuator line approach is O(100) faster than resolving all geometry

Conclusions

- Developed a first verified version of a block-based dynamically adaptive LBM for wind turbine wake aerodynamics using actuator models (disc model not shown here)
- Actuator line approach is O(100) faster than resolving all geometry
- Vorticity generation along blades noticably different

Conclusions

- Developed a first verified version of a block-based dynamically adaptive LBM for wind turbine wake aerodynamics using actuator models (disc model not shown here)
- Actuator line approach is O(100) faster than resolving all geometry
- Vorticity generation along blades noticably different
- Thanks to the low dissipation property of the LBM wake convection behavior is excellent. Hierachical meshes are vital for efficiency.

Conclusions

- Developed a first verified version of a block-based dynamically adaptive LBM for wind turbine wake aerodynamics using actuator models (disc model not shown here)
- Actuator line approach is O(100) faster than resolving all geometry
- Vorticity generation along blades noticably different
- Thanks to the low dissipation property of the LBM wake convection behavior is excellent. Hierachical meshes are vital for efficiency.
- Currently testing more complex LES turbulence models: dynamic Smagorinsky and wall-adaptive linear eddy models

Conclusions

- Developed a first verified version of a block-based dynamically adaptive LBM for wind turbine wake aerodynamics using actuator models (disc model not shown here)
- Actuator line approach is O(100) faster than resolving all geometry
- Vorticity generation along blades noticably different
- Thanks to the low dissipation property of the LBM wake convection behavior is excellent. Hierachical meshes are vital for efficiency.
- Currently testing more complex LES turbulence models: dynamic Smagorinsky and wall-adaptive linear eddy models
- Should test also different source term incorporations, cf. [Mohamad and Kuzmin, 2010], for actuator models.

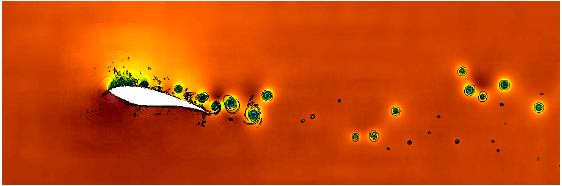
Conclusions

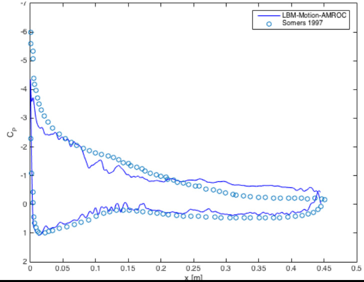
- Developed a first verified version of a block-based dynamically adaptive LBM for wind turbine wake aerodynamics using actuator models (disc model not shown here)
- Actuator line approach is O(100) faster than resolving all geometry
- Vorticity generation along blades noticably different
- Thanks to the low dissipation property of the LBM wake convection behavior is excellent. Hierachical meshes are vital for efficiency.
- Currently testing more complex LES turbulence models: dynamic Smagorinsky and wall-adaptive linear eddy models
- Should test also different source term incorporations, cf. [Mohamad and Kuzmin, 2010], for actuator models.
- Consideration of tower and ground topology, realistic inflow conditions in the near future.

Outlook

- For accurate prediction of shear flows and boundary layers, a wall-function model for high Re flows will be implemented.

S825 airfoil – $\alpha = 13.1^\circ$, $\mathrm{Re} = 2 \cdot 10^6$





References I

[Bouzidi et al., 2001] Bouzidi, M., Firdaouss, M., and Lallemand, P. (2001). Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids*, 13:3452.

[Chen et al., 2006] Chen, H., Filippova, O., Hoch, J., Molvig, K., Shock, R., Teixeira, C., and Zhang, R. (2006). Grid refinement in lattice Boltzmann methods based on volumetric formulation. *Physica A*, 362:158–167.

[Deiterding, 2011] Deiterding, R. (2011). Block-structured adaptive mesh refinement - theory, implementation and application. *European Series in Applied and Industrial Mathematics: Proceedings*, 34:97–150.

[Deiterding et al., 2007] Deiterding, R., Cirak, F., Mauch, S. P., and Meiron, D. I. (2007). A virtual test facility for simulating detonation- and shock-induced deformation and fracture of thin flexible shells. *Int. J. Multiscale Computational Engineering*, 5(1):47–63.

[Deiterding et al., 2006] Deiterding, R., Radovitzky, R., Mauch, S. P., Noels, L., Cummings, J. C., and Meiron, D. I. (2006). A virtual test facility for the efficient simulation of solid materials under high energy shock-wave loading. *Engineering with Computers*, 22(3-4):325–347.

[Deiterding and Wood, 2015] Deiterding, R. and Wood, S. L. (2015). A dynamically adaptive lattice boltzmann method for predicting wake phenomena in fully coupled wind engineering problems. In Schrefler, B., Onate, E., and Papadrakakis, M., editors, *IV Int. Conf. on Coupled Problems in Science and Engineering*, pages 489–500.

[Deiterding and Wood, 2016a] Deiterding, R. and Wood, S. L. (2016a). An adaptive lattice boltzmann method for predicting wake fields behind wind turbines. In Dillmann, A. e. a., editor, *New Results in Numerical and Experimental Fluid Mechanics X*, volume 132 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 845–857. Springer.

[Deiterding and Wood, 2016b] Deiterding, R. and Wood, S. L. (2016b). Predictive wind turbine simulation with an adaptive lattice boltzmann method for moving boundaries. *J. Phys. Conf. Series*, 753:082005.

[Henderson, 1995] Henderson, R. D. (1995). Details of the drag curve near the onset of vortex shedding. *Phys. Fluids*, 7:2102–2104.

[Hou et al., 1996] Hou, S., Sterling, J., Chen, S., and Doolen, G. D. (1996). A lattice Boltzmann subgrid model for high Reynolds number flows. In Lawniczak, A. T. and Kapral, R., editors, *Pattern formation and lattice gas automata*, volume 6, pages 151–166. Fields Inst Comm.

[Mauch, 2000] Mauch, S. (2000). A fast algorithm for computing the closest point and distance transform. *SIAM J. Scientific Comput.*

[Mohamad and Kuzmin, 2010] Mohamad, A. A. and Kuzmin, A. (2010). A critical evaluation of force term in lattice Boltzmann method, natural convection problem. *International Journal of Heat and Mass Transfer*, 53(5-6):990–996.

References II

[Schlaffer, 2013] Schlaffer, M. B. (2013). *Non-reflecting boundary conditions for the lattice Boltzmann method*. PhD thesis, Technical University Munich.

[Shen et al., 2005] Shen, W. Z., Mikkelsen, R., Sørensen, J. N., and Bak, C. (2005). Tip loss corrections for wind turbine computations. *Wind Energy*, 8(4):457–475.

[Sørensen et al., 1998] Sørensen, J. N., Shen, W. Z., and Munduate, X. (1998). Analysis of wake states by a full-field actuator disc model. *Wind Energy*, 1(2):73–88.

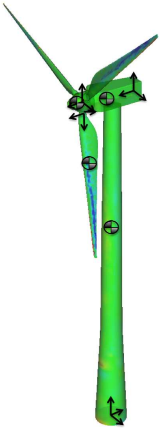
[Tsai, 1999] Tsai, L. (1999). *Robot Analysis: The Mechanics of Serial and Parallel Manipulators*. Wiley.

[Yu, 2004] Yu, H. (2004). *Lattice Boltzmann equation simulations of turbulence, mixing, and combustion*. PhD thesis, Texas A&M University.

Motion solver

Based on the Newton-Euler method solution of dynamics equation of kinetic chains [Tsai, 1999]

$$\begin{pmatrix} \mathbf{F} \\ \boldsymbol{\tau}_{\mathrm{P}} \end{pmatrix} = \begin{pmatrix} m\mathbf{1} & -m[\mathbf{c}]^{\times} \\ m[\mathbf{c}]^{\times}\mathbf{I}_{\mathrm{cm}} & -m[\mathbf{c}]^{\times}[\mathbf{c}]^{\times} \end{pmatrix} \begin{pmatrix} \mathbf{a}_{\mathrm{P}} \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} m[\boldsymbol{\omega}]^{\times}[\boldsymbol{\omega}]^{\times}\mathbf{c} \\ [\boldsymbol{\omega}]^{\times}(\mathbf{I}_{\mathrm{cm}} - m[\mathbf{c}]^{\times}[\mathbf{c}]^{\times})\,\boldsymbol{\omega} \end{pmatrix}.$$

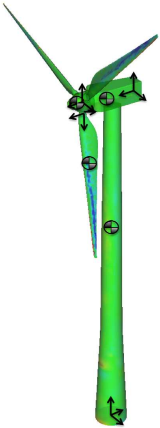


m = mass of the body, 1 = the 4×4 homogeneous identity matrix,
$\mathbf{a}_p$ = acceleration of link frame with origin at $\mathbf{p}$ in the preceding link's frame,
$\mathbf{I}_{\mathrm{cm}}$ = moment of inertia about the center of mass,
$\boldsymbol{\omega}$ = angular velocity of the body,
$\boldsymbol{\alpha}$ = angular acceleration of the body,
$\mathbf{c}$ is the location of the body's center of mass,
and $[\mathbf{c}]^{\times}$, $[\boldsymbol{\omega}]^{\times}$ denote skew-symmetric cross product matrices.

Motion solver

Based on the Newton-Euler method solution of dynamics equation of kinetic chains [Tsai, 1999]

$$\begin{pmatrix} \mathbf{F} \\ \boldsymbol{\tau}_{\mathrm{P}} \end{pmatrix} = \begin{pmatrix} m\mathbf{1} & -m[\mathbf{c}]^{\times} \\ m[\mathbf{c}]^{\times}\mathbf{I}_{\mathrm{cm}} & -m[\mathbf{c}]^{\times}[\mathbf{c}]^{\times} \end{pmatrix} \begin{pmatrix} \mathbf{a}_{\mathrm{P}} \\ \boldsymbol{\alpha} \end{pmatrix} + \begin{pmatrix} m[\boldsymbol{\omega}]^{\times}[\boldsymbol{\omega}]^{\times}\mathbf{c} \\ [\boldsymbol{\omega}]^{\times}(\mathbf{I}_{\mathrm{cm}} - m[\mathbf{c}]^{\times}[\mathbf{c}]^{\times})\,\boldsymbol{\omega} \end{pmatrix}.$$



m = mass of the body, 1 = the 4×4 homogeneous identity matrix,
$\mathbf{a}_p$ = acceleration of link frame with origin at $\mathbf{p}$ in the preceding link's frame,
$\mathbf{I}_{\mathrm{cm}}$ = moment of inertia about the center of mass,
$\boldsymbol{\omega}$ = angular velocity of the body,
$\boldsymbol{\alpha}$ = angular acceleration of the body,
$\mathbf{c}$ is the location of the body's center of mass,
and $[\mathbf{c}]^{\times}$, $[\boldsymbol{\omega}]^{\times}$ denote skew-symmetric cross product matrices.

Here, we additionally define the total force and torque acting on a body,

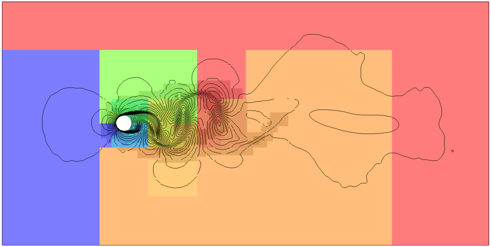
$\mathbf{F} = (\mathbf{F}_{FSI} + \mathbf{F}_{prescribed}) \cdot \boldsymbol{\mathcal{C}}_{xyz}$ and

$\tau = (\tau_{FSI} + \tau_{prescribed}) \cdot \boldsymbol{\mathcal{C}}_{\alpha\beta\gamma}$ respectively.

Where $\boldsymbol{\mathcal{C}}_{xyz}$ and $\boldsymbol{\mathcal{C}}_{\alpha\beta\gamma}$ are the translational and rotational constraints, respectively.

Flow over 2D cylinder, $d = 2\,\mathrm{cm}$

- Air with $\nu = 1.61 \cdot 10^{-5}\,\mathrm{m}^2/\mathrm{s}$, $\rho = 1.205\,\mathrm{kg}/\mathrm{m}^3$
- Domain size $[-8d, 24d] \times [-8d, 8d]$
- Dynamic refinement based on velocity. Last level to refine structure further.
- Inflow from left. Characteristic boundary conditions [Schlaffer, 2013] elsewhere.



- Base lattice 320×160, 3 additional levels with factors $2, 4, 4$.
- Resolution: ~ 320 points in diameter d
- Computation of C_D on 400 equidistant points along circle and averaged over time. Comparison above with [Henderson, 1995].

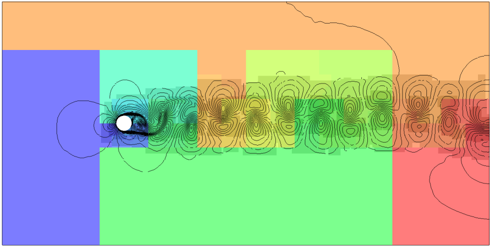
Flow over 2D cylinder, $d = 2\,\mathrm{cm}$

- Air with $\nu = 1.61 \cdot 10^{-5}\,\mathrm{m}^2/\mathrm{s}$, $\rho = 1.205\,\mathrm{kg}/\mathrm{m}^3$
- Domain size $[-8d, 24d] \times [-8d, 8d]$
- Dynamic refinement based on velocity. Last level to refine structure further.
- Inflow from left. Characteristic boundary conditions [Schlaffer, 2013] elsewhere.

- Base lattice 320×160, 3 additional levels with factors $2, 4, 4$.
- Resolution: ~ 320 points in diameter d
- Computation of C_D on 400 equidistant points along circle and averaged over time. Comparison above with [Henderson, 1995].

Flow over 2D cylinder, $d = 2\,\mathrm{cm}$

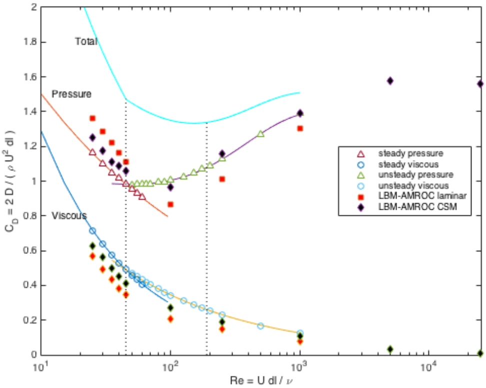
- Air with $\nu = 1.61 \cdot 10^{-5}\,\mathrm{m}^2/\mathrm{s}$, $\rho = 1.205\,\mathrm{kg}/\mathrm{m}^3$
- Domain size $[-8d, 24d] \times [-8d, 8d]$
- Dynamic refinement based on velocity. Last level to refine structure further.
- Inflow from left. Characteristic boundary conditions [Schlaffer, 2013] elsewhere.



- Base lattice 320×160, 3 additional levels with factors $2, 4, 4$.
- Resolution: ~ 320 points in diameter d
- Computation of C_D on 400 equidistant points along circle and averaged over time. Comparison above with [Henderson, 1995].

Flow over 2D cylinder, $d = 2\,\mathrm{cm}$

- Air with $\nu = 1.61 \cdot 10^{-5}\,\mathrm{m}^2/\mathrm{s}$, $\rho = 1.205\,\mathrm{kg}/\mathrm{m}^3$
- Domain size $[-8d, 24d] \times [-8d, 8d]$
- Dynamic refinement based on velocity. Last level to refine structure further.
- Inflow from left. Characteristic boundary conditions [Schlaffer, 2013] elsewhere.



- Base lattice 320×160, 3 additional levels with factors $2, 4, 4$.
- Resolution: ~ 320 points in diameter d
- Computation of C_D on 400 equidistant points along circle and averaged over time. Comparison above with [Henderson, 1995].