Aerodynamics and fluid-structure interaction simulation with AMROC Part II

Ralf Deiterding

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

Xiamen
24th July, 2019

Outline

Approximation of Boltzmann equation

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

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

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

Approximation of Boltzmann equation

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

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

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

Equation is approximated with a splitting approach.

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

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

Approximation of Boltzmann equation

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

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

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

Equation is approximated with a splitting approach.

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

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

$$\rho(\mathbf{x}, t) = \sum_{\alpha=0}^{8} f_\alpha(\mathbf{x}, t), \quad \rho(\mathbf{x}, t) u_i(\mathbf{x}, t) = \sum_{\alpha=0}^{8} \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x}, t)$$

Approximation of Boltzmann equation

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

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

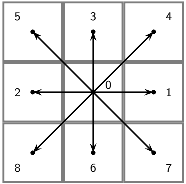
- $\mathrm{Kn} = l_f / L \ll 1$, where l_f is replaced with Δx
- Weak compressibilty and small Mach number assumed
- Assume a simplified phase space

Equation is approximated with a splitting approach.

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

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

$$\rho(\mathbf{x}, t) = \sum_{\alpha=0}^{8} f_\alpha(\mathbf{x}, t), \quad \rho(\mathbf{x}, t) u_i(\mathbf{x}, t) = \sum_{\alpha=0}^{8} \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x}, t)$$



Discrete velocities:

$\mathbf{e}_0 = (0, 0), \mathbf{e}_1 = (1, 0)c, \mathbf{e}_2 = (-1, 0)c, \mathbf{e}_3 = (0, 1)c, \mathbf{e}_4 = (1, 1)c, \ldots$

Approximation of Boltzmann equation

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

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

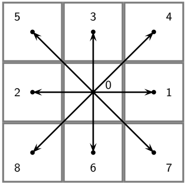
- $\mathrm{Kn} = l_f / L \ll 1$, where l_f is replaced with Δx
- Weak compressibilty and small Mach number assumed
- Assume a simplified phase space

Equation is approximated with a splitting approach.

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

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

$$\rho(\mathbf{x}, t) = \sum_{\alpha=0}^{8} f_\alpha(\mathbf{x}, t), \quad \rho(\mathbf{x}, t) u_i(\mathbf{x}, t) = \sum_{\alpha=0}^{8} \mathbf{e}_{\alpha i} f_\alpha(\mathbf{x}, t)$$



Discrete velocities:

$\mathbf{e}_0 = (0,0), \mathbf{e}_1 = (1,0)c, \mathbf{e}_2 = (-1,0)c, \mathbf{e}_3 = (0,1)c, \mathbf{e}_4 = (1,1)c, \ldots$

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

Approximation of Boltzmann equation

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

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

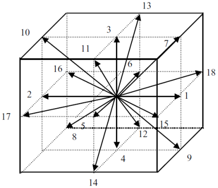
- $\mathrm{Kn} = l_f / L \ll 1$, where l_f is replaced with Δx
- Weak compressibilty and small Mach number assumed
- Assume a simplified phase space

Equation is approximated 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:

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

Approximation of equilibrium state

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

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

Approximation of equilibrium state

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

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

with equilibrium function

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

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

Approximation of equilibrium state

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

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

with equilibrium function

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

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 $\Sigma_{ij} = \left(1 - \frac{\omega_L \Delta t}{2}\right) \sum_\alpha \mathbf{e}_{\alpha i} \mathbf{e}_{\alpha j} (f_\alpha^{eq} - f_\alpha)$

Approximation of equilibrium state

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

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

with equilibrium function

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

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 $\Sigma_{ij} = \left(1 - \frac{\omega_L \Delta t}{2}\right) \sum_\alpha \mathbf{e}_{\alpha i} \mathbf{e}_{\alpha j} (f_\alpha^{eq} - f_\alpha)$
Is derived by assuming a Maxwell-Boltzmann distribution of f_α^{eq} and approximating the involved exp() function with a Taylor series to second-order accuracy.

Approximation of equilibrium state

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

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

with equilibrium function

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

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 $\Sigma_{ij} = \left(1 - \frac{\omega_L \Delta t}{2}\right) \sum_\alpha \mathbf{e}_{\alpha i} \mathbf{e}_{\alpha j} (f_\alpha^{eq} - f_\alpha)$
Is derived by assuming a Maxwell-Boltzmann distribution of f_α^{eq} and approximating the involved exp() function with a Taylor series to second-order accuracy.
Using the third-order equilibrium function

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

allows higher flow velocities.

Relation to Navier-Stokes equations

Inserting a Chapman-Enskog expansion, that is,

$$f_\alpha = f_\alpha(0) + \epsilon f_\alpha(1) + \epsilon^2 f_\alpha(2) + ...$$

and using

$$\frac{\partial}{\partial t} = \epsilon \frac{\partial}{\partial t_1} + \epsilon^2 \frac{\partial}{\partial t_2} + ..., \qquad \nabla = \epsilon \nabla_1 + \epsilon^2 \nabla_2 + ...$$

into the LBM and summing over α one can show that the continuity and moment equations are recoverd to $O(\epsilon^2)$ [Hou et al., 1996]

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}$$

Relation to Navier-Stokes equations

Inserting a Chapman-Enskog expansion, that is,

$$f_\alpha = f_\alpha(0) + \epsilon f_\alpha(1) + \epsilon^2 f_\alpha(2) + \ldots$$

and using

$$\frac{\partial}{\partial t} = \epsilon \frac{\partial}{\partial t_1} + \epsilon^2 \frac{\partial}{\partial t_2} + \ldots, \qquad \nabla = \epsilon \nabla_1 + \epsilon^2 \nabla_2 + \ldots$$

into the LBM and summing over α one can show that the continuity and moment equations are recoverd to $O(\epsilon^2)$ [Hou et al., 1996]

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u}$$

Kinematic viscosity and collision time are connected by

$$\nu = \frac{1}{3} \left(\frac{\tau_L}{\Delta t} - \frac{1}{2} \right) c \Delta x$$

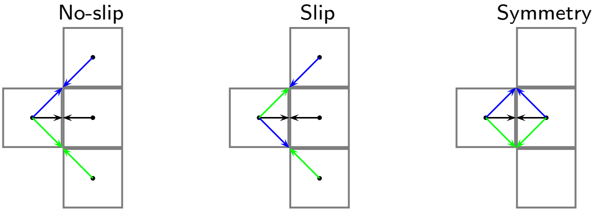
from which one gets with $\sqrt{3} c_s = \frac{\Delta x}{\Delta t}$ [Hähnel, 2004]

$$\omega_L = \tau_L^{-1} = \frac{c_s^2}{\nu + \Delta t c_s^2 / 2}$$

Initial and boundary conditions

- Initial conditions are constructed as $f_\alpha^{eq}(\rho(t=0), \mathbf{u}(t=0))$

Boundary conditions (applied before streaming step)

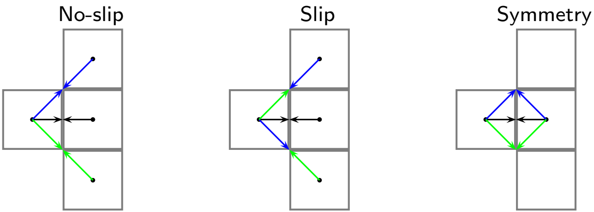


- Outlet basically copies all distributions (zero gradient)
- Inlet and pressure boundary conditions use f_α^{eq}

Initial and boundary conditions

- Initial conditions are constructed as $f_\alpha^{eq}(\rho(t=0), \mathbf{u}(t=0))$

Boundary conditions (applied before streaming step)



- Outlet basically copies all distributions (zero gradient)
- Inlet and pressure boundary conditions use f_α^{eq}

Complex geometry:

- Use level set method as before to construct macro-values in embedded boundary cells by interpolation / extrapolation [Deiterding, 2011].
- Distance function φ, normal $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$. Triangulated meshes use CPT algorithm [Mauch, 2003].
- Construct macro-velocity in ghost cells for no-slip BC as $\mathbf{u}' = 2\mathbf{w} - \mathbf{u}$
- Then use $f_\alpha^{eq}(\rho', \mathbf{u}')$ or interpolated bounce-back [Bouzidi et al., 2001] to construct distributions in embedded ghost cells

Normalization

The method is implemented on the unit lattice with $\Delta\tilde{x} = \Delta\tilde{t} = 1$

$$\frac{\Delta x}{l_0} = 1, \quad \frac{\Delta t}{t_0} = 1 \longrightarrow c = 1$$

Normalization

The method is implemented on the unit lattice with $\Delta\tilde{x} = \Delta\tilde{t} = 1$

$$\frac{\Delta x}{l_0} = 1, \quad \frac{\Delta t}{t_0} = 1 \longrightarrow c = 1$$

Lattice viscosity $\tilde{\nu} = \frac{1}{3}\left(\tau - \frac{1}{2}\right)$ and lattice sound speed $\tilde{c}_s = \frac{1}{\sqrt{3}}$ yield again

$$\omega_L = \frac{\tilde{c}_s^2}{\nu' + \tilde{c}_s^2/2} = \frac{c_s^2}{\nu + \Delta t c_s^2/2}$$

Normalization

The method is implemented on the unit lattice with $\Delta\tilde{x} = \Delta\tilde{t} = 1$

$$\frac{\Delta x}{l_0} = 1, \quad \frac{\Delta t}{t_0} = 1 \longrightarrow c = 1$$

Lattice viscosity $\tilde{\nu} = \frac{1}{3}\left(\tau - \frac{1}{2}\right)$ and lattice sound speed $\tilde{c}_s = \frac{1}{\sqrt{3}}$ yield again

$$\omega_L = \frac{\tilde{c}_s^2}{\nu' + \tilde{c}_s^2/2} = \frac{c_s^2}{\nu + \Delta t c_s^2/2}$$

Velocity normalization factor: $u_0 = \frac{l_0}{t_0}$, density ρ_0

$$\mathrm{Re} = \frac{uL}{\nu} = \frac{u/u_0 \cdot l/l_0}{\nu/(u_0 l_0)} = \frac{\tilde{u}\tilde{l}}{\tilde{\nu}}$$

Normalization

The method is implemented on the unit lattice with $\Delta\tilde{x} = \Delta\tilde{t} = 1$

$$\frac{\Delta x}{l_0} = 1, \quad \frac{\Delta t}{t_0} = 1 \longrightarrow c = 1$$

Lattice viscosity $\tilde{\nu} = \frac{1}{3}\left(\tau - \frac{1}{2}\right)$ and lattice sound speed $\tilde{c}_s = \frac{1}{\sqrt{3}}$ yield again

$$\omega_L = \frac{\tilde{c}_s^2}{\nu' + \tilde{c}_s^2/2} = \frac{c_s^2}{\nu + \Delta t c_s^2/2}$$

Velocity normalization factor: $u_0 = \frac{l_0}{t_0}$, density ρ_0

$$\mathrm{Re} = \frac{uL}{\nu} = \frac{u/u_0 \cdot l/l_0}{\nu/(u_0 l_0)} = \frac{\tilde{u}\tilde{l}}{\tilde{\nu}}$$

Trick for scheme acceleration: Use $\bar{u} = Su$ and $\bar{\nu} = S\nu$ which yields

$$\bar{\omega}_L = \frac{c_s^2}{S\nu + \Delta t/S\, c_s^2/2}$$

Normalization

The method is implemented on the unit lattice with $\Delta\tilde{x} = \Delta\tilde{t} = 1$

$$\frac{\Delta x}{l_0} = 1, \quad \frac{\Delta t}{t_0} = 1 \longrightarrow c = 1$$

Lattice viscosity $\tilde{\nu} = \frac{1}{3}\left(\tau - \frac{1}{2}\right)$ and lattice sound speed $\tilde{c}_s = \frac{1}{\sqrt{3}}$ yield again

$$\omega_L = \frac{\tilde{c}_s^2}{\nu' + \tilde{c}_s^2/2} = \frac{c_s^2}{\nu + \Delta t c_s^2/2}$$

Velocity normalization factor: $u_0 = \frac{l_0}{t_0}$, density ρ_0

$$\mathrm{Re} = \frac{uL}{\nu} = \frac{u/u_0 \cdot l/l_0}{\nu/(u_0 l_0)} = \frac{\tilde{u}\tilde{l}}{\tilde{\nu}}$$

Trick for scheme acceleration: Use $\bar{u} = Su$ and $\bar{\nu} = S\nu$ which yields

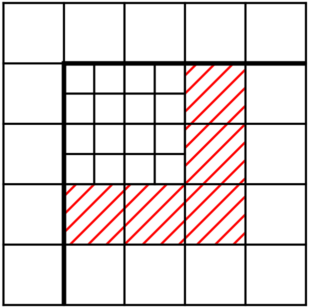
$$\bar{\omega}_L = \frac{c_s^2}{S\nu + \Delta t/S\, c_s^2/2}$$

For instance, the physical hydrodynamic pressure is then obtained for a caloric gas as

$$p = (\tilde{\rho} - 1)\tilde{c}_s^2 \frac{u_0^2}{S^2}\rho_0 + \frac{c_s^2 \rho_0}{\gamma}$$

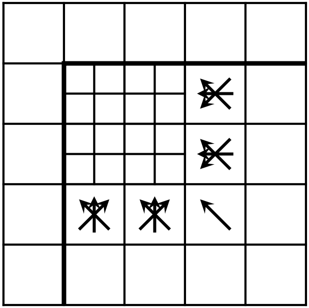
Adaptive LBM

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



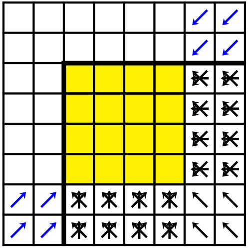
Adaptive LBM

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

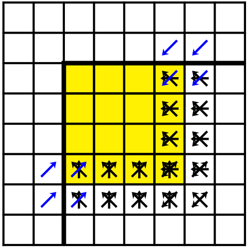
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.



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

Adaptive LBM

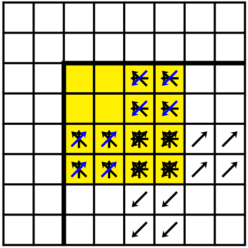
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2}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



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

Adaptive LBM

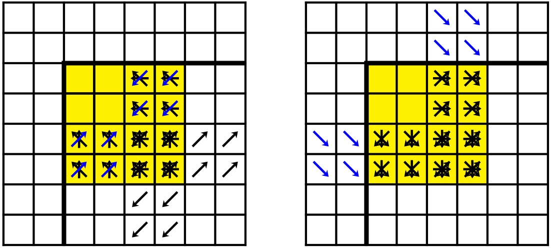
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2}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

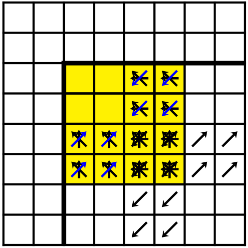
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2}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



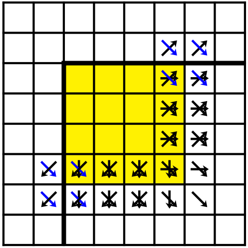
$\tilde{f}_{\alpha,in}^{f,n+1/2}$ $\qquad$ $f_{\alpha,out}^{f,n}$

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.



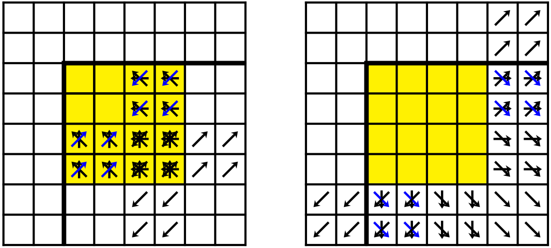
$\tilde{f}_{\alpha,in}^{f,n+1/2}$



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

Adaptive LBM

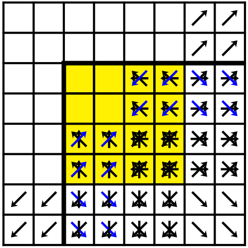
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2}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



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

Adaptive LBM

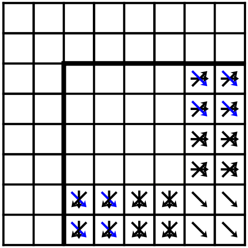
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2}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

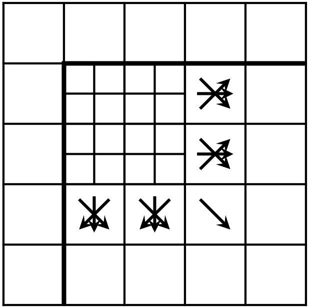


$\tilde{f}_{\alpha,out}^{f,n+1/2}$, $\tilde{f}_{\alpha,out}^{f,n}$

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

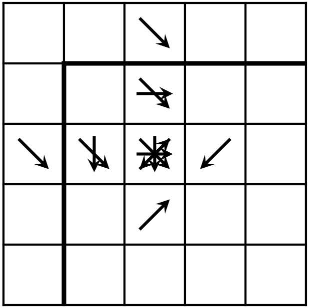
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2}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

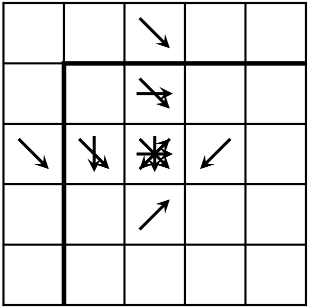
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2}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

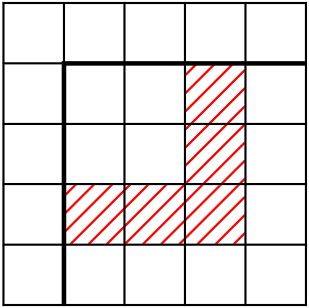
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2}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

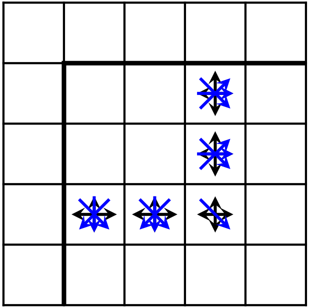
1. Complete update on coarse grid: $f_\alpha^{C,n+1} := \mathcal{CT}(f_\alpha^{C,n})$
2. Interpolate $f_{\alpha,in}^{C,n}$ onto $f_{\alpha,in}^{f,n}$ to fill fine halos. Set physical boundary conditions.
3. $\tilde{f}_\alpha^{f,n} := \mathcal{T}(f_\alpha^{f,n})$ on whole fine mesh. $f_\alpha^{f,n+1/2}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

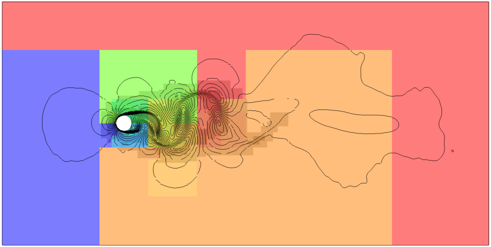


5. Average $\tilde{f}_{\alpha,out}^{f,n+1/2}$ (inner halo layer), $\tilde{f}_{\alpha,out}^{f,n}$ (outer halo layer) to obtain $\tilde{f}_{\alpha,out}^{C,n}$.
6. Revert transport into halos: $\bar{f}_{\alpha,out}^{C,n} := \mathcal{T}^{-1}(\tilde{f}_{\alpha,out}^{C,n})$
7. Parallel synchronization of $f_\alpha^{C,n},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

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].

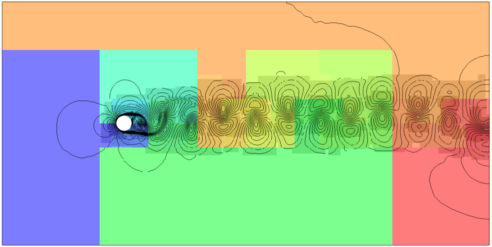
Flow over 2D cylinder, $d = 2\,\mathrm{cm}$

- 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, $d = 2\,\mathrm{cm}$

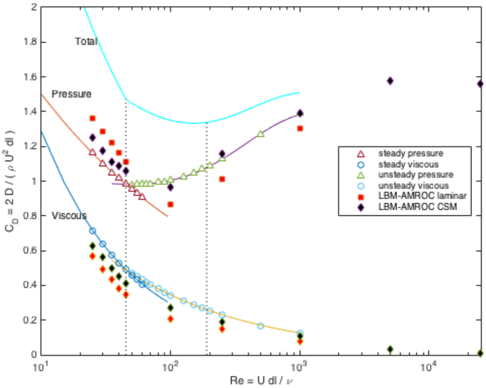
- Air with $\nu = 1.61 \cdot 10^{-5}\,\mathrm{m^2/s}$, $\rho = 1.205\,\mathrm{kg/m^3}$
- Domain size $[-8d, 24d] \times [-8d, 8d]$
- Dynamic refinement based on velocity. Last level to refine structure further.
- Inflow from left. Characteristic boundary conditions [Schlaffer, 2013] elsewhere.



- 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, $d = 2\,\mathrm{cm}$

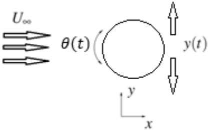
- Air with $\nu = 1.61 \cdot 10^{-5}\,\mathrm{m}^2/\mathrm{s}$, $\rho = 1.205\,\mathrm{kg}/\mathrm{m}^3$
- Domain size $[-8d, 24d] \times [-8d, 8d]$
- Dynamic refinement based on velocity. Last level to refine structure further.
- Inflow from left. Characteristic boundary conditions [Schlaffer, 2013]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

Oscillating cylinder – Setup

Motion imposed on cylinder



Case	A_t	$f_t = f_\theta$	V_R	U_∞	Re
1a	$D/4$	0.6	0.5	0.0606	1322
1b	$D/2$	0.6	1.0	0.0606	1322
2a	$D/4$	3.0	0.5	0.3030	6310
2b	$D/2$	3.0	1.0	0.3030	6310

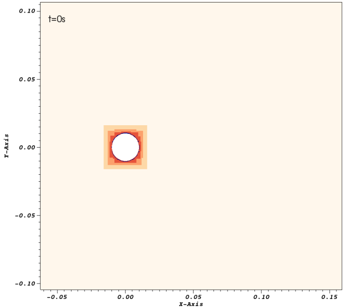
$$y(t) = A_t \sin(2\pi f_t t), \qquad \theta(t) = A_\theta \sin(2\pi f_\theta t)$$

- Setup follows [Nazarinia et al., 2012]. Here $A_\theta = 1$ for all cases.
- Natural frequency of cylinder $f_N \approx 0.6154\,\mathrm{s}^{-1}$.
- Strouhal number $\mathrm{St}_t = f_t D/U_\infty \approx 0.198$ for all cases.
- Chosen here $D = 20\,\mathrm{mm}$
- Fluid is water with $c_s = 1482\,\mathrm{m/s}$, $\nu = 9.167 \cdot 10^{-7}\,\mathrm{m^2/s}$, $\rho = 1016\,\mathrm{kg/m^3}$
- Constant coefficient model deactivated for Case 1, active for Case 2 with $C_{sm} = 0.2$

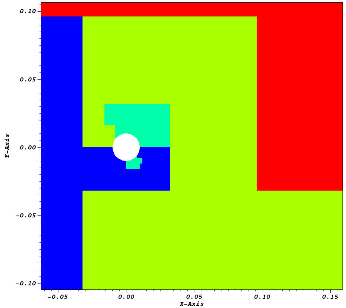
C. Laloglu, RD. Proc. 5th Int. Conf. on Parallel, Distributed, Grid and Cloud Computing for Engineering, Civil-Comp Press, 2017.

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



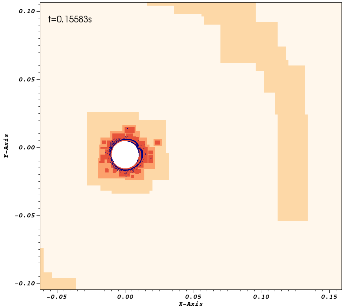
Distribution to 4 processors



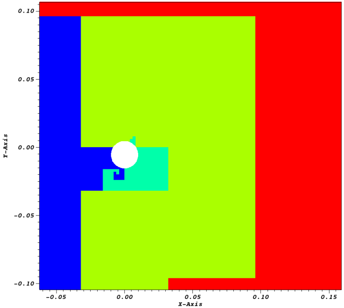
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



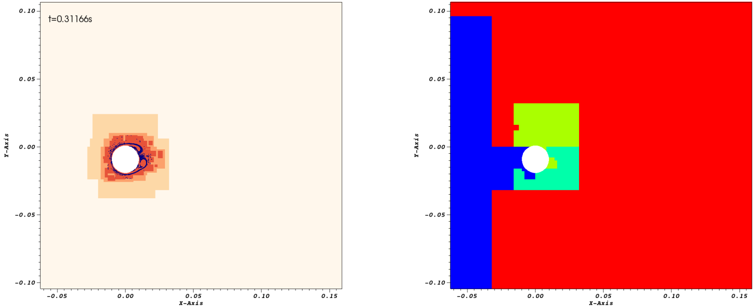
Distribution to 4 processors



- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

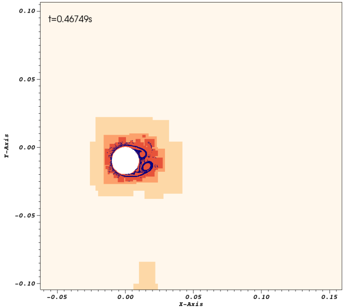
Mesh refinement Distribution to 4 processors



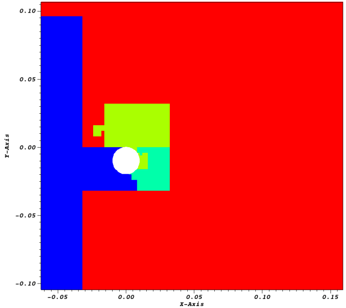
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



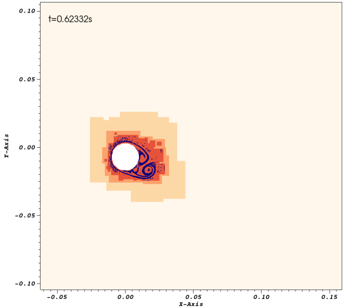
Distribution to 4 processors



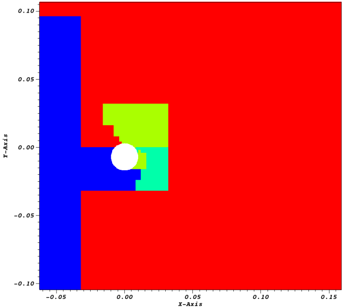
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



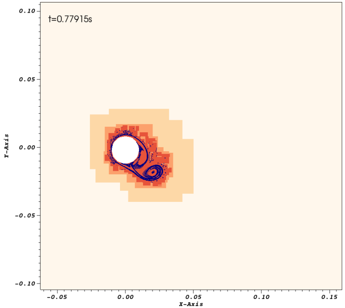
Distribution to 4 processors



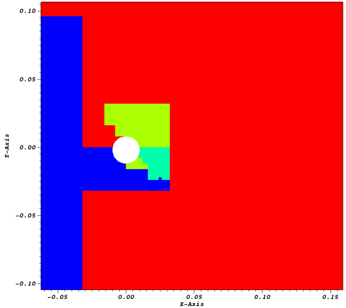
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



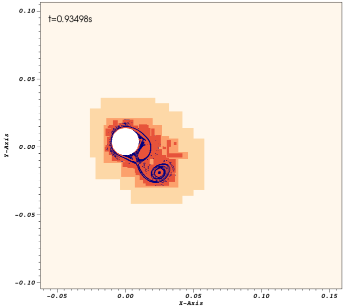
Distribution to 4 processors



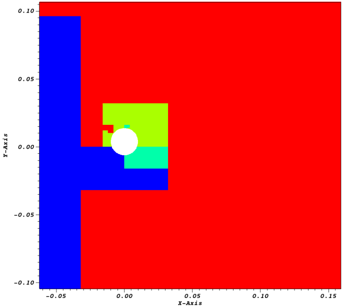
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



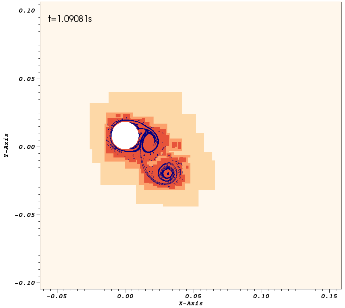
Distribution to 4 processors



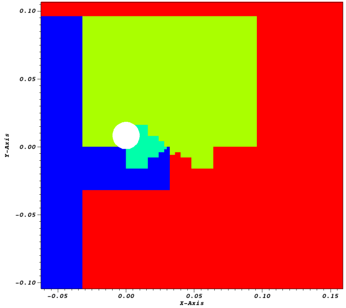
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



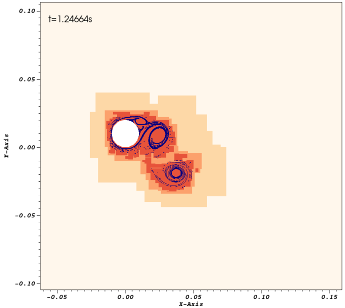
Distribution to 4 processors



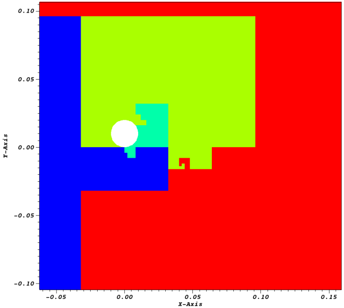
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



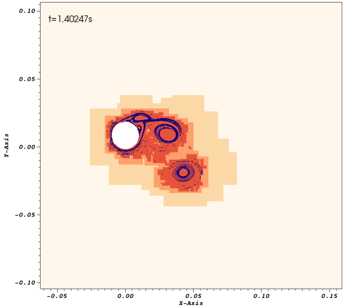
Distribution to 4 processors



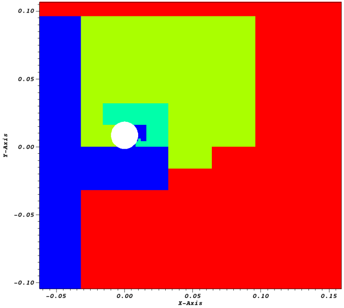
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



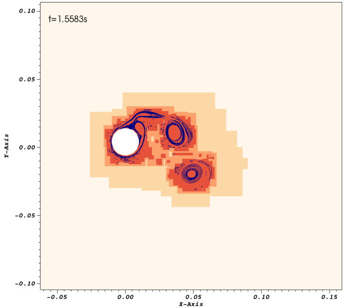
Distribution to 4 processors



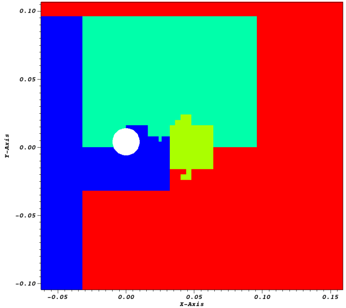
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



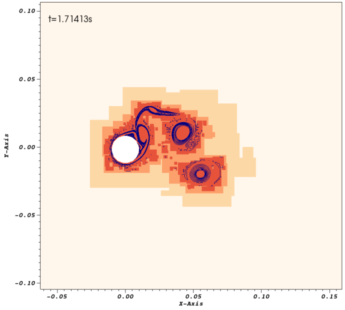
Distribution to 4 processors



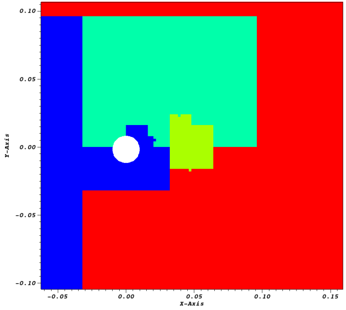
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



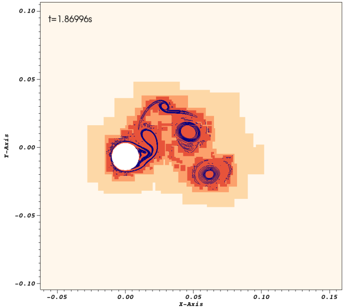
Distribution to 4 processors



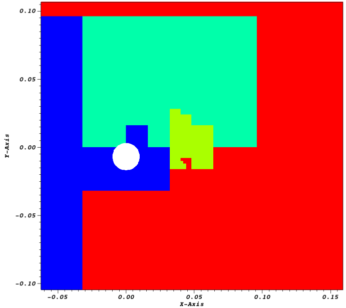
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



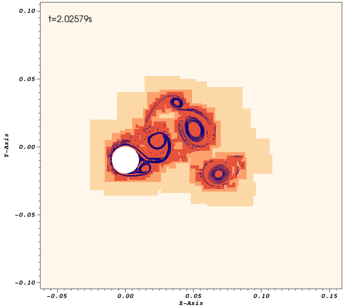
Distribution to 4 processors



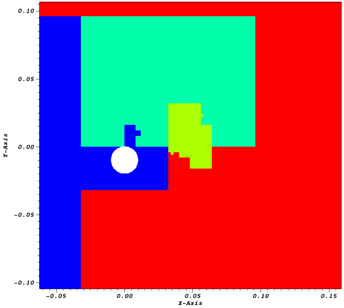
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



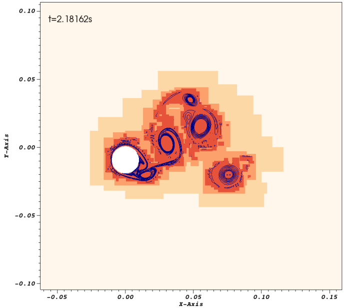
Distribution to 4 processors



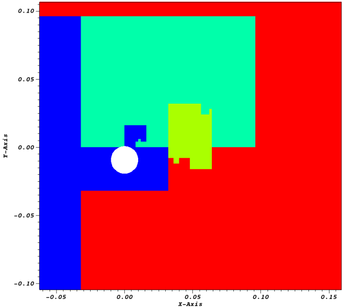
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



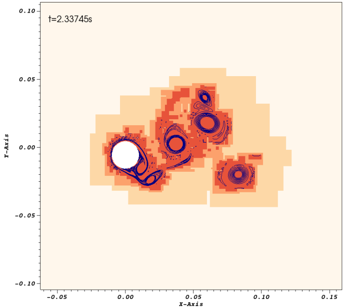
Distribution to 4 processors



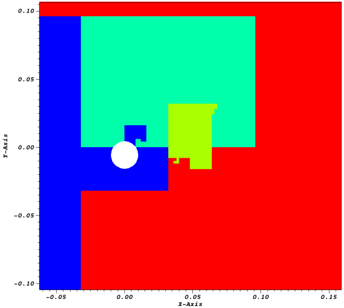
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



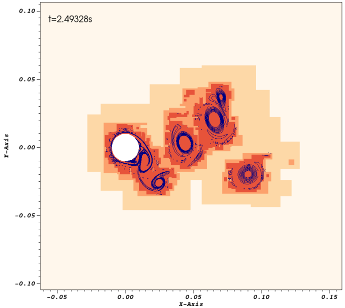
Distribution to 4 processors



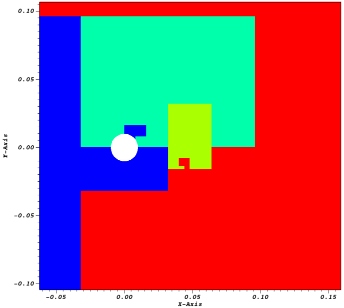
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



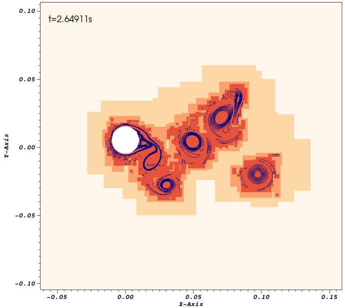
Distribution to 4 processors



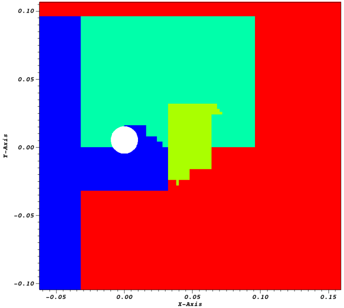
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



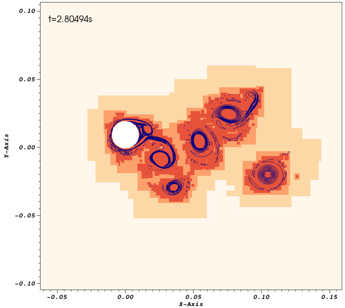
Distribution to 4 processors



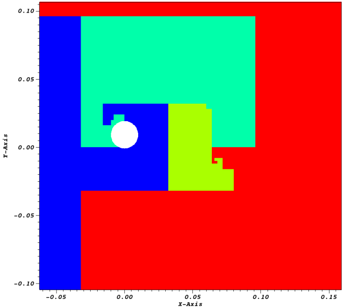
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



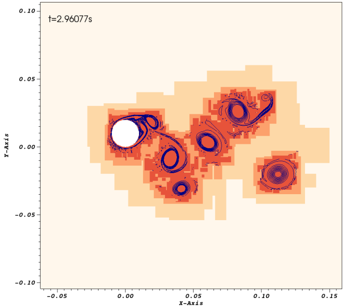
Distribution to 4 processors



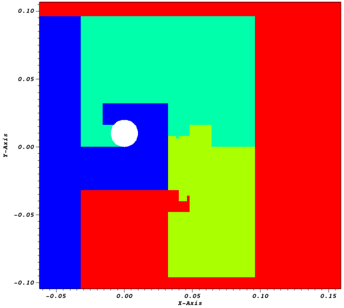
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



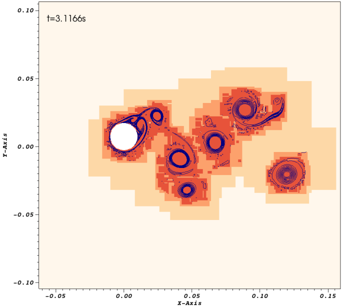
Distribution to 4 processors



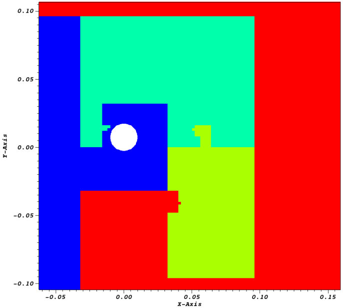
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



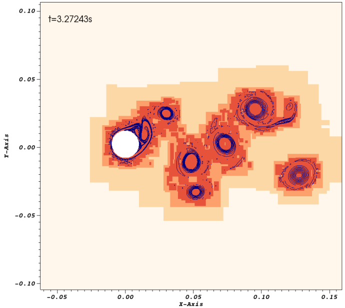
Distribution to 4 processors



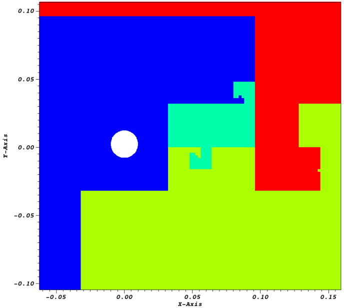
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



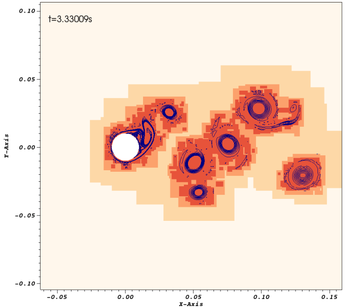
Distribution to 4 processors



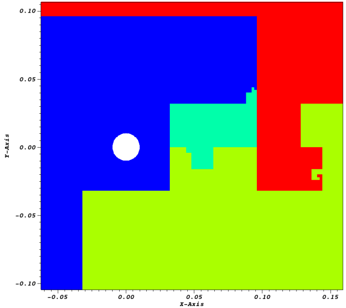
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



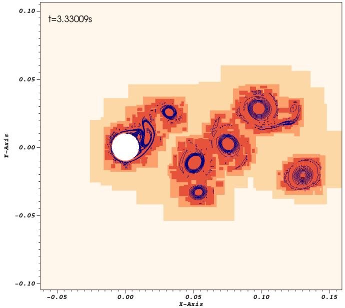
Distribution to 4 processors



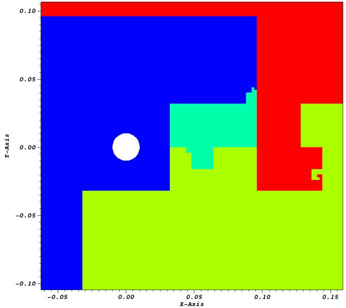
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$

Case 1b, $V_R = 1$, $\mathrm{Re} = 1322$

Mesh refinement



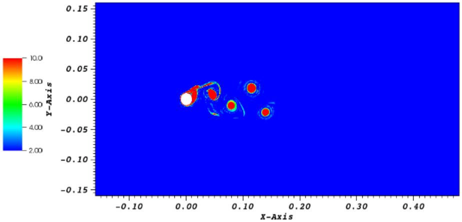
Distribution to 4 processors



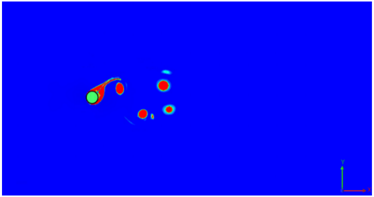
- Visualization enlargement of cylinder region
- Base mesh is discretized with 320×160 cells, 3 additional levels with factor $r_l = 2, 2, 2$
- 80 cells within D on highest level
- Speedup $S = 2000$
- Basically identical setup in commercial code XFlow for comparison

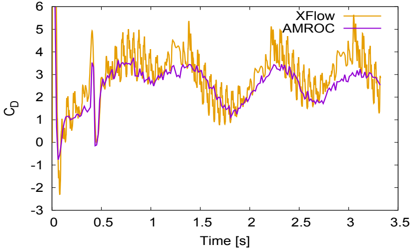
Case 1b, $V_R = 1$, $f_t = f_\theta = 0.6$, $\mathrm{Re} = 1322$

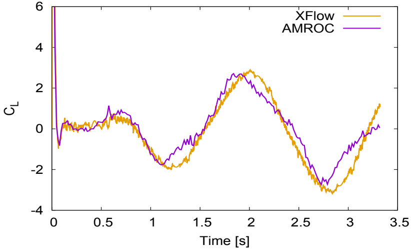
AMROC



XFlow

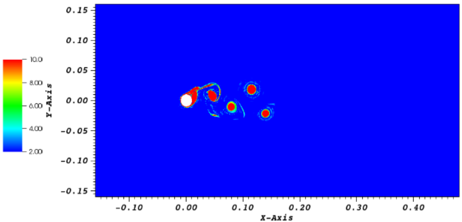




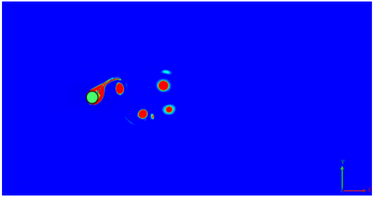


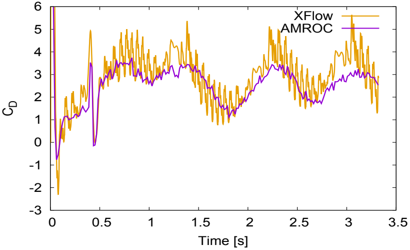
Case 1b, $V_R = 1$, $f_t = f_\theta = 0.6$, $\mathrm{Re} = 1322$

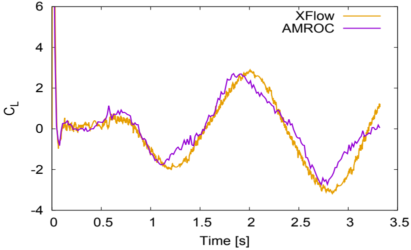
AMROC



XFlow



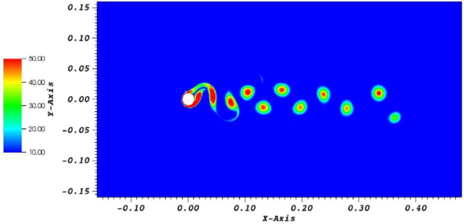




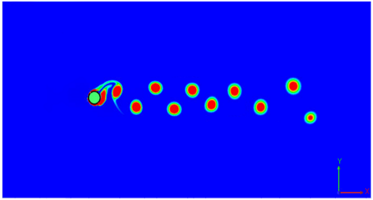
- Increase of rotational velocity leads to formation of a vortex pair plus single vortex. Drag and lift amplitude roughly doubled.
- Laminar results in good agreement with experiments of [Nazarinia et al., 2012].

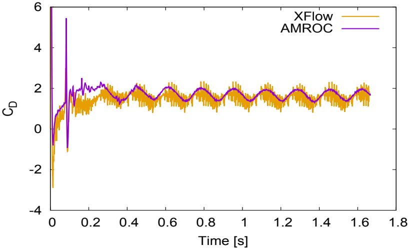
Case 2a, $V_R = 0.5$, $f_t = f_\theta = 3$, $\mathrm{Re} = 6310$

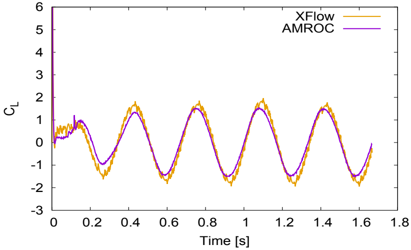
AMROC



XFlow

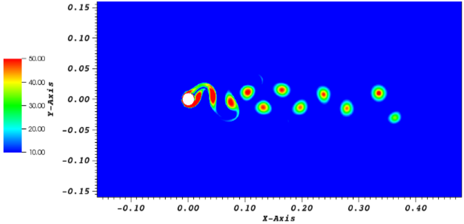




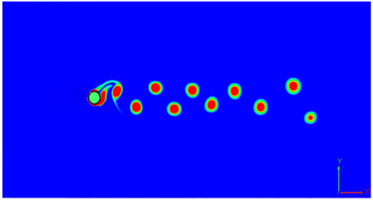


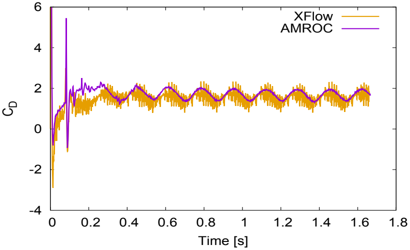
Case 2a, $V_R = 0.5$, $f_t = f_\theta = 3$, $\mathrm{Re} = 6310$

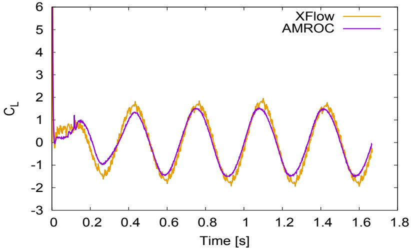
AMROC



XFlow







- Oscillation period: $T = 1/f_t = 0.33\,\mathrm{s}$. 10 regular vortices in $1.67\,\mathrm{s}$.
- CPU time on 6 cores for AMROC: 635.8 s, XFlow $\sim 50\,\%$ more expensive when normalized based on number of cells

Computational performance

Flow type	Case	Δt_0 [s]	Total cells		Δt_e [s]	Re	y^+	CPU time [s]	
			AMROC	XFlow				AMROC	XFlow
Laminar	**1a**	0.0015	85982	84778	3.33	1322	0	161.89	176
	1b	0.0015	91774	90488	3.33	1322	0	165.97	183
Turbulent	**2a**	0.00031	232840	216452	1.66	6310	2.4	635.8	887
	2b	0.00031	255582	246366	1.66	6310	2.6	933.2	1325

- Intel-Xeon-3.50-GHz desktop workstation with 6 cores, communication through MPI
- Same base mesh and always three additional refinement levels
- AMROC: single-relaxation time LBM, block-based mesh adaptation
- XFlow: slightly more multi-relaxation time LBM, cell-based mesh adaptation

Computational performance

Flow type	Case	Δt_0 [s]	Total cells		Δt_e [s]	Re	y^+	CPU time [s]	
			AMROC	XFlow				AMROC	XFlow
Laminar	**1a**	0.0015	85982	84778	3.33	1322	0	161.89	176
	1b	0.0015	91774	90488	3.33	1322	0	165.97	183
Turbulent	**2a**	0.00031	232840	216452	1.66	6310	2.4	635.8	887
	2b	0.00031	255582	246366	1.66	6310	2.6	933.2	1325

- Intel-Xeon-3.50-GHz desktop workstation with 6 cores, communication through MPI
- Same base mesh and always three additional refinement levels
- AMROC: single-relaxation time LBM, block-based mesh adaptation
- XFlow: slightly more multi-relaxation time LBM, cell-based mesh adaptation
- AMROC uses $\sim 7.5\,\%$ more cells on average more cells
- Normalized on cell number Case 2a is 50 % more expensive for XFlow than for AMROC-LBM
- Case 2b is 42 % more expensive in CPU time alone

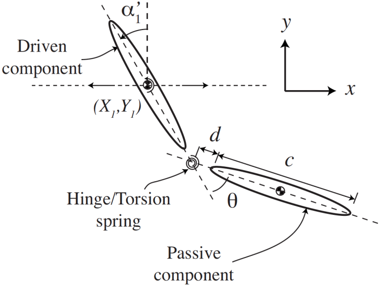
Two-segment hinged wing

Configuration by [Toomey and Eldredge, 2008].
Manufactured bodies in tank filled with water.
Prescribed translation and rotation

$$X_t(t) = \frac{A_0}{2}\frac{G_t(ft)}{max\ Gt}C(ft), \quad \alpha_1(t) = -\beta\frac{G_r(ft)}{max\ Gr}$$

with $G_r(t) = tanh[\sigma_r cos(2\pi t + \Phi)]$,

$$G_t(t) = \int_t tanh[\sigma_t cos(2\pi t')]dt'$$



$A_0\ (cm)$	7.1
$c\ (cm)$	5.1
$d\ (cm)$	0.25
$\rho_b\ (\mathrm{kg/m^3})$	5080
$f\ (Hz)$	0.15

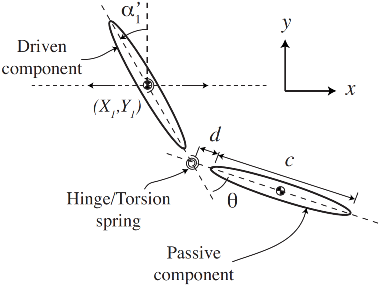
Two-segment hinged wing

Configuration by [Toomey and Eldredge, 2008].
Manufactured bodies in tank filled with water.
Prescribed translation and rotation

$$X_t(t) = \frac{A_0}{2}\frac{G_t(ft)}{max\ Gt}C(ft), \quad \alpha_1(t) = -\beta\frac{G_r(ft)}{max\ Gr}$$

with $G_r(t) = tanh[\sigma_r cos(2\pi t + \Phi)]$,

$$G_t(t) = \int_t tanh[\sigma_t cos(2\pi t')]dt'$$



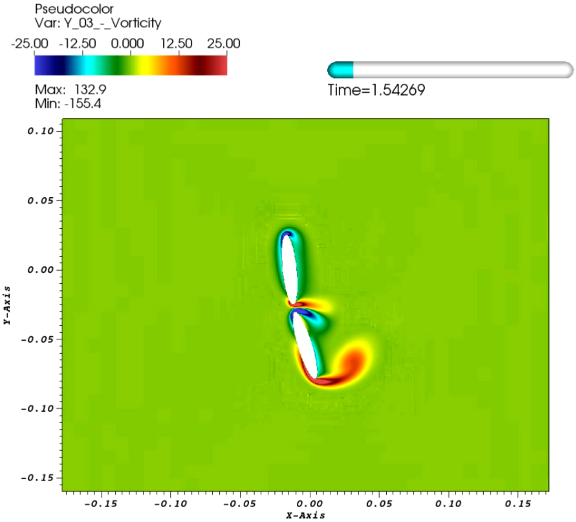
- 7 cases constructed by varying σ_r, σ_t, Φ
- Rotational Reynolds number $\mathrm{Re_r} = 2\pi\beta\sigma_r fc^2/(\tanh(\sigma_r)\nu)$ varied between 2200 and 7200 in experiments
- [Toomey and Eldredge, 2008] reference simulations with a viscous particle method are for $\mathrm{Re_r} = \{100, 500\}$

$A_0\ (cm)$	7.1
$c\ (cm)$	5.1
$d\ (cm)$	0.25
$\rho_b\ (\mathrm{kg/m^3})$	5080
$f\ (Hz)$	0.15

Case 1 - $\sigma_r = \sigma_t = 0.628$, $\Phi = 0$, $\mathrm{Re}_r = 100$

- Quiescent water
  $\rho_f = 997\,\mathrm{kg/m^3}$
  $c_s = 1497\,\mathrm{m/s}$
- No-slip boundaries in y, periodic in x-direction
- Base level: 100×100 for $[-0.5, 0.5] \times [-0.5, 0.5]$ domain
- 4 additional levels with factors 2,2,2,4
- Coupling to rigid body motion solver on 4th level

Right: computed vorticity field (enlarged)



Quantitative comparison

- Evaluate normalized force $F_{x,y} = 2F^*_{x,y}/(\rho_f^2 c^3)$ and moment $M = 2M^*/(\rho_f f^2 c^4)$ over 3 periods
- [Wood and Deiterding, 2015] Used finest spatial resolution $\Delta x/c = 0.0122$
  [Toomey and Eldredge, 2008]: $\Delta x/c = 0.013$ ($\mathrm{Re_r} = 100$), $\Delta x/c = 0.0032$ ($\mathrm{Re_r} = 500$)
- Temporal resolution ~ 113 and ~ 28 times finer

Relative difference in mean force and moment

	$\mathrm{Re_r} = 100$			$\mathrm{Re_r} = 500$		
Case	$\bar{F}_x$	$\bar{F}_y$	$\bar{M}$	$\bar{F}_x$	$\bar{F}_y$	$\bar{M}$
1	-2.59	3.33	-3.85	3.33	5.45	-3.75
2	2.47	0.74	2.55	2.35	3.83	-4.29
3	1.27	0.45	0.72	2.31	4.65	-3.43
4	4.86	4.28	3.54	3.51	2.37	-2.32
5	4.83	0.47	0.25	4.34	4.39	-2.67
6	2.10	3.19	1.52	3.00	1.82	-3.96
7	1.41	0.99	3.28	4.31	2.32	-3.07

Relative difference in peak force and moment

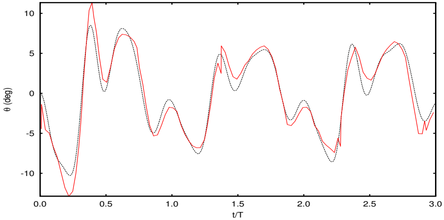
	$\mathrm{Re_r} = 100$			$\mathrm{Re_r} = 500$		
Case	$\lvert\lvert F_x\rvert\rvert_\infty$	$\lvert\lvert F_y\rvert\rvert_\infty$	$\lvert\lvert M\rvert\rvert_\infty$	$\lvert\lvert F_x\rvert\rvert_\infty$	$\lvert\lvert F_y\rvert\rvert_\infty$	$\lvert\lvert M\rvert\rvert_\infty$
1	4.40	5.07	-3.66	4.40	3.98	-4.17
2	4.46	2.42	2.62	2.72	4.33	-2.34
3	4.20	3.20	4.80	3.32	2.68	-4.59
4	4.67	2.22	3.71	0.18	2.51	-2.85
5	3.57	3.37	1.26	4.09	4.97	-3.63
6	2.04	3.08	1.52	3.92	2.08	-4.44
7	2.20	1.91	2.26	3.29	3.79	-4.40

Quantitative comparison

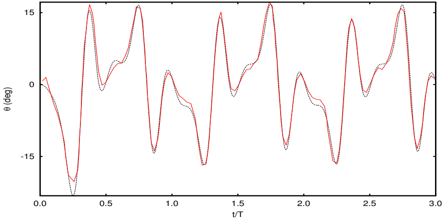
- Evaluate normalized force $F_{x,y} = 2F^*_{x,y}/(\rho_f^2 c^3)$ and moment $M = 2M^*/(\rho_f f^2 c^4)$ over 3 periods
- [Wood and Deiterding, 2015] Used finest spatial resolution $\Delta x/c = 0.0122$
  [Toomey and Eldredge, 2008]: $\Delta x/c = 0.013$ ($\mathrm{Re_r} = 100$), $\Delta x/c = 0.0032$ ($\mathrm{Re_r} = 500$)
- Temporal resolution ~ 113 and ~ 28 times finer

Hinge deflection angle over time

Case 1
$\sigma_t = 0.628$
$\sigma_r = 0.628$
$\Phi = 0$



Case 2
$\sigma_t = 1.885$
$\sigma_r = 1.885$
$\Phi = 0^o$



Experimental results (–); Current (- -)

An LBM for thermal transport

Consider the Navier-Stokes equations under Boussinesq approximation

$$\nabla \cdot \mathbf{u} = 0$$
$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F}$$
$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u} T) = \mathcal{D} \nabla^2 T$$

with $\mathbf{F} = \mathbf{g}\beta\,(T - T_{ref})$.

An LBM for thermal transport

Consider the Navier-Stokes equations under Boussinesq approximation

$$\nabla \cdot \mathbf{u} = 0$$
$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F}$$
$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u} T) = \mathcal{D} \nabla^2 T$$

with $\mathbf{F} = \mathbf{g}\beta \left(T - T_{ref} \right)$.

An LBM for this system needs to use two distribution functions f_α and g_α.

1.) Transport step $\mathcal{T}$:

$$\tilde{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t), \quad \tilde{g}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = g_\alpha(\mathbf{x}, t)$$

An LBM for thermal transport

Consider the Navier-Stokes equations under Boussinesq approximation

$$\nabla \cdot \mathbf{u} = 0$$
$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{F}$$
$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u} T) = \mathcal{D} \nabla^2 T$$

with $\mathbf{F} = \mathbf{g}\beta\,(T - T_{ref})$.

An LBM for this system needs to use two distribution functions f_α and g_α.

1.) Transport step $\mathcal{T}$:

$$\tilde{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t), \quad \tilde{g}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = g_\alpha(\mathbf{x}, t)$$

2.) Collision step $\mathcal{C}$:

$$f_\alpha(\cdot, t + \Delta t) = \tilde{f}_\alpha(\cdot, t + \Delta t) + \omega_{L,\nu} \Delta t \left(\tilde{f}_\alpha^{eq}(\cdot, t + \Delta t) - \tilde{f}_\alpha(\cdot, t + \Delta t) \right) + \Delta t \mathbf{F}_\alpha$$

$$g_\alpha(\cdot, t + \Delta t) = \tilde{g}_\alpha(\cdot, t + \Delta t) + \omega_{L,\mathcal{D}} \Delta t \left(\tilde{g}_\alpha^{eq}(\cdot, t + \Delta t) - \tilde{g}_\alpha(\cdot, t + \Delta t) \right)$$

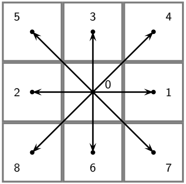
with collision frequencies

$$\omega_{L,\nu} = \frac{c_s^2}{\nu + c_s^2 \Delta t/2}, \quad \omega_{L,\mathcal{D}} = \frac{\frac{3}{2} c_s^2}{\mathcal{D} + \frac{3}{2} c_s^2 \Delta t/2}$$

Equilibrium operators

This incompressible method uses in 2D [Guo et al., 2002]

$$f_\alpha^{(eq)} = \begin{cases} -4\sigma_0 p - s_\alpha(\mathbf{u}), & \text{for } \alpha = 0, \\ \sigma_\alpha p + s_\alpha(\mathbf{u}), & \text{for } \alpha = 1, \ldots, 8, \end{cases}$$



where

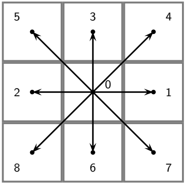
$$s_\alpha(\mathbf{u}) = t_\alpha \left[\frac{3\mathbf{e}_\alpha \mathbf{u}}{c^2} + \frac{9(\mathbf{e}_\alpha \mathbf{u})^2}{2c^4} - \frac{3\mathbf{u}^2}{2c^2} \right]$$

with $t_\alpha = \frac{1}{9}\left\{4, 1, 1, 1, \frac{1}{4}, \frac{1}{4}, 1, \frac{1}{4}, \frac{1}{4}\right\}$ and $\sigma_\alpha = \frac{1}{3}\left\{-5, 1, 1, 1, \frac{1}{4}, \frac{1}{4}, 1, \frac{1}{4}, \frac{1}{4}\right\}$

Equilibrium operators

This incompressible method uses in 2D [Guo et al., 2002]

$$f_\alpha^{(eq)} = \begin{cases} -4\sigma_0 p - s_\alpha(\mathbf{u}), & \text{for } \alpha = 0, \\ \sigma_\alpha p + s_\alpha(\mathbf{u}), & \text{for } \alpha = 1, \ldots, 8, \end{cases}$$



where

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

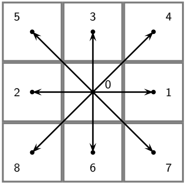
with $t_\alpha = \frac{1}{9}\left\{4, 1, 1, 1, \frac{1}{4}, \frac{1}{4}, 1, \frac{1}{4}, \frac{1}{4}\right\}$ and $\sigma_\alpha = \frac{1}{3}\left\{-5, 1, 1, 1, \frac{1}{4}, \frac{1}{4}, 1, \frac{1}{4}, \frac{1}{4}\right\}$

$$g_\alpha^{(eq)} = \frac{T}{4}\left[1 + 2\mathbf{e}_\alpha \cdot \mathbf{u}\right] \quad \text{for } \alpha = 1, \ldots, 4$$

Equilibrium operators

This incompressible method uses in 2D [Guo et al., 2002]

$$f_\alpha^{(eq)} = \begin{cases} -4\sigma_0 p - s_\alpha(\mathbf{u}), & \text{for } \alpha = 0, \\ \sigma_\alpha p + s_\alpha(\mathbf{u}), & \text{for } \alpha = 1, \dots, 8, \end{cases}$$



where

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

with $t_\alpha = \frac{1}{9}\left\{4, 1, 1, 1, \frac{1}{4}, \frac{1}{4}, 1, \frac{1}{4}, \frac{1}{4}\right\}$ and $\sigma_\alpha = \frac{1}{3}\left\{-5, 1, 1, 1, \frac{1}{4}, \frac{1}{4}, 1, \frac{1}{4}, \frac{1}{4}\right\}$

$$g_\alpha^{(eq)} = \frac{T}{4}\left[1 + 2\mathbf{e}_\alpha \cdot \mathbf{u}\right] \quad \text{for } \alpha = 1, \dots, 4$$

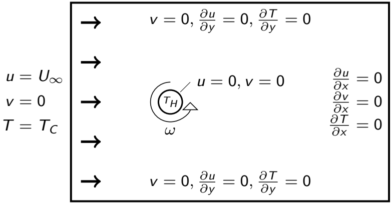
Forces are applied in y-direction only:

$$F_\alpha = \frac{1}{2}\left(\delta_{i3} - \delta_{i6}\right)\mathbf{e}_i \cdot \mathbf{F}$$

Moments: $\mathbf{u} = \sum_{\alpha>0} \mathbf{e}_i f_\alpha, \quad p = \frac{1}{4\sigma}\left[\sum_{\alpha>0} f_\alpha + s_0(\mathbf{u})\right], \quad T = \sum_{\alpha=1}^{4} g_\alpha$

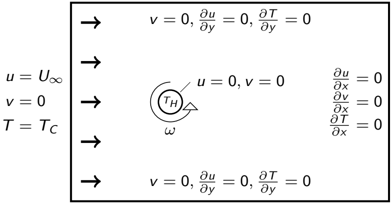
Heated rotating cylinder

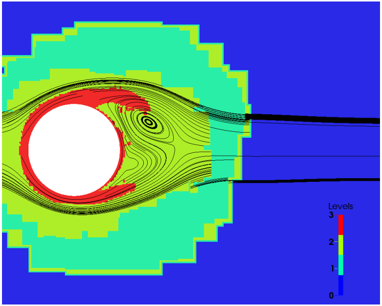
- $R = 15$, domain: $[-6R, 16R] \times [-8R, 8R]$
- $\mathrm{Re} = 2U_\infty R/\nu = 200$, $U_\infty = 0.01$
- Peripheral velocity $V = \Omega R$, $V/U_\infty = 0.5$
- Base grid 288×240 refined by three levels with $r_1 = 2$, $r_{2,3} = 4$ using scaled gradients of u, v, T



Heated rotating cylinder

- $R = 15$, domain: $[-6R, 16R] \times [-8R, 8R]$
- $\mathrm{Re} = 2U_\infty R/\nu = 200$, $U_\infty = 0.01$
- Peripheral velocity $V = \Omega R$, $V/U_\infty = 0.5$
- Base grid 288×240 refined by three levels with $r_1 = 2$, $r_{2,3} = 4$ using scaled gradients of u, v, T

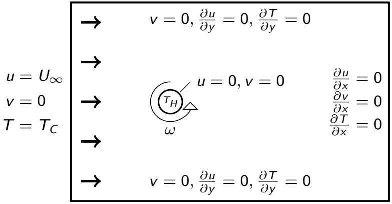


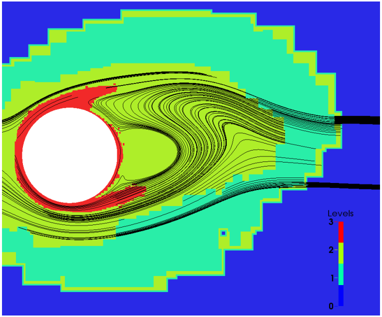


$t = 3$

Heated rotating cylinder

- $R = 15$, domain: $[-6R, 16R] \times [-8R, 8R]$
- $\mathrm{Re} = 2U_\infty R/\nu = 200$, $U_\infty = 0.01$
- Peripheral velocity $V = \Omega R$, $V/U_\infty = 0.5$
- Base grid 288×240 refined by three levels with $r_1 = 2$, $r_{2,3} = 4$ using scaled gradients of u, v, T

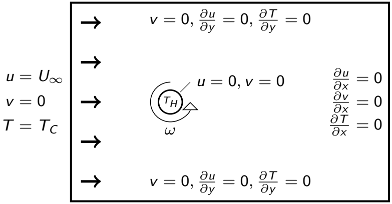


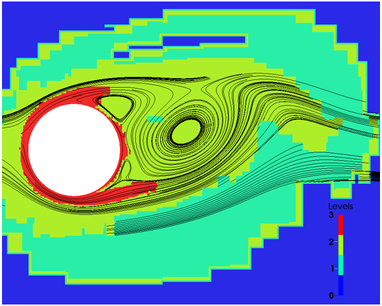


$t = 6$

Heated rotating cylinder

- $R = 15$, domain: $[-6R, 16R] \times [-8R, 8R]$
- $\mathrm{Re} = 2U_\infty R/\nu = 200$, $U_\infty = 0.01$
- Peripheral velocity $V = \Omega R$, $V/U_\infty = 0.5$
- Base grid 288×240 refined by three levels with $r_1 = 2$, $r_{2,3} = 4$ using scaled gradients of u, v, T

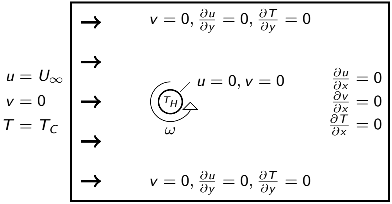


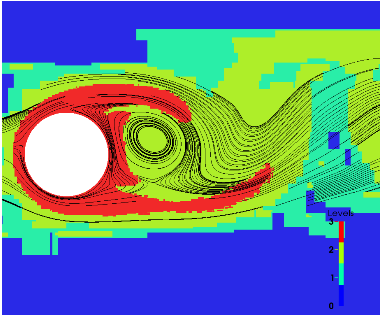


$t = 8$

Heated rotating cylinder

- $R = 15$, domain: $[-6R, 16R] \times [-8R, 8R]$
- $\mathrm{Re} = 2U_\infty R/\nu = 200$, $U_\infty = 0.01$
- Peripheral velocity $V = \Omega R$, $V/U_\infty = 0.5$
- Base grid 288×240 refined by three levels with $r_1 = 2$, $r_{2,3} = 4$ using scaled gradients of u, v, T

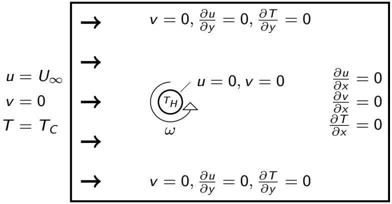


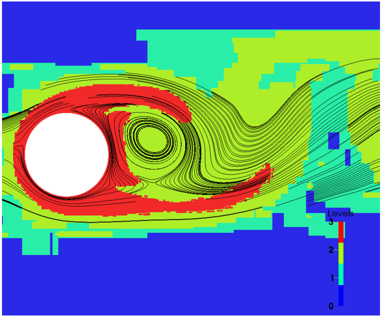


$t = 12$

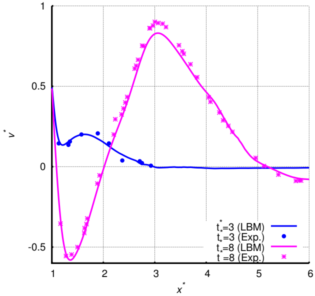
Heated rotating cylinder

- $R = 15$, domain: $[-6R, 16R] \times [-8R, 8R]$
- $\mathrm{Re} = 2U_\infty R/\nu = 200$, $U_\infty = 0.01$
- Peripheral velocity $V = \Omega R$, $V/U_\infty = 0.5$
- Base grid 288×240 refined by three levels with $r_1 = 2$, $r_{2,3} = 4$ using scaled gradients of u, v, T





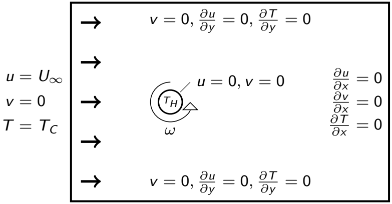
$t = 12$

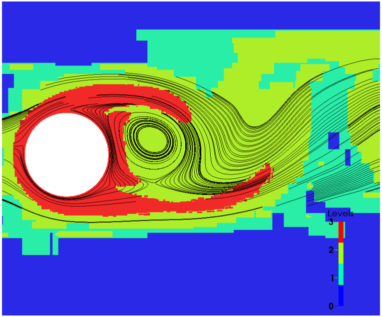


v velocity along x-axis

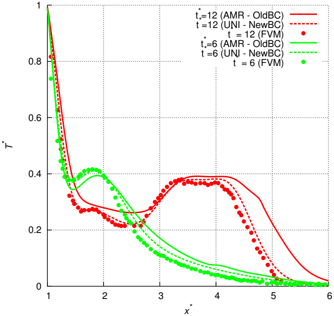
Heated rotating cylinder

- $R = 15$, domain: $[-6R, 16R] \times [-8R, 8R]$
- $\mathrm{Re} = 2U_\infty R/\nu = 200$, $U_\infty = 0.01$
- Peripheral velocity $V = \Omega R$, $V/U_\infty = 0.5$
- Base grid 288×240 refined by three levels with $r_1 = 2$, $r_{2,3} = 4$ using scaled gradients of u, v, T





$t = 12$



Temperature T along x-axis

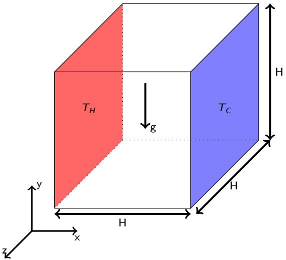
Natural convection

Characterized by

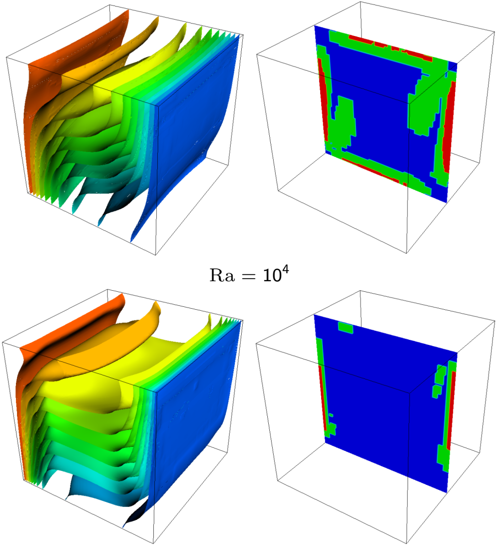
$$\mathrm{Ra} = \frac{g\beta\Delta T H^3}{\nu\mathcal{D}}$$

a = AMROC-LBM,
b = [Fusegi et al., 1991] (NS - uniform)

	Ref.	$u_{\max}$	$y_{\max}$	$v_{\max}$	$x_{\max}$	$\mathrm{Nu}_{\mathrm{ave}}$
$\mathrm{Ra} = 10^3$	a	0.132	0.195	0.132	0.829	1.099
	b	0.131	0.200	0.132	0.833	1.105
$\mathrm{Ra} = 10^4$	a	0.197	0.194	0.220	0.887	2.270
	b	0.201	0.183	0.225	0.883	2.302
$\mathrm{Ra} = 10^5$	a	0.141	0.152	0.242	0.935	4.583
	b	0.147	0.145	0.247	0.935	4.646



Isosurfaces of temperature and refinement levels



$\mathrm{Ra} = 10^4$

$\mathrm{Ra} = 10^5$

K. Feldhusen, RD, C. Wagner. J. Applied Math. Comp. Science 26(4): 735–747, 2016.

Turbulence modeling

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

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

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

Turbulence modeling

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

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

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

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

Turbulence modeling

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

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

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

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., $\nu_t = (C_{sm}\Delta x)^2 |\overline{\mathbf{S}}|$, where

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

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

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

Turbulence modeling

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

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

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

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

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

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

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

$$\bar{S}_{ij} = \frac{\bar{\Sigma}_{ij}}{2\rho c_s^2 \tau^\star_L \left(1 - \frac{\omega_L \Delta t}{2}\right)} = \frac{1}{2\rho c_s^2 \tau^\star_L}\sum_\alpha e_{\alpha i} e_{\alpha j}(\bar{f}^{eq}_\alpha - \bar{f}_\alpha)$$

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

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

Further LES models

Dynamic Smagorinsky model (DSMA)

$$C_{sm}(\mathbf{x},t)^2 = -\frac{1}{2}\frac{\langle L_{ij} M_{ij}\rangle}{\langle M_{ij} M_{ij}\rangle}$$

$$L_{ij} = T_{ij} - \widehat{\tau}_{ij} = \widehat{\overline{u}_i\overline{u}_j} - \hat{\overline{u}}_i\hat{\overline{u}}_j \qquad M_{ij} = \widehat{\Delta x}^2 |\hat{\overline{\mathbf{S}}}|\hat{\overline{S}}_{ij} - \Delta x^2 \widehat{|\overline{\mathbf{S}}|\overline{S}_{ij}}$$

No van Driest damping implemented yet!

Further LES models

Dynamic Smagorinsky model (DSMA)

$$C_{sm}(\mathbf{x},t)^2 = -\frac{1}{2}\frac{\langle L_{ij}M_{ij}\rangle}{\langle M_{ij}M_{ij}\rangle}$$

$$L_{ij} = T_{ij} - \widehat{\tau}_{ij} = \widehat{\overline{u}_i\overline{u}_j} - \hat{\overline{u}}_i\hat{\overline{u}}_j \qquad M_{ij} = \widehat{\Delta x}^2|\hat{\overline{\mathbf{S}}}|\hat{\overline{S}}_{ij} - \Delta x^2\widehat{|\overline{\mathbf{S}}|\overline{S}_{ij}}$$

No van Driest damping implemented yet!

Wall-Adapting Local Eddy-viscosity model (WALE)

$$\nu_t = (C_w\Delta x)^2 OP_{WALE}, \quad \text{where } C_w = 0.5$$

WALE turbulence time-scale

$$OP_{WALE} = \frac{(\mathcal{J}_{ij}\mathcal{J}_{ij})^{\frac{3}{2}}}{(\overline{S}_{ij}\overline{S}_{ij})^{\frac{5}{2}} + (\mathcal{J}_{ij}\mathcal{J}_{ij})^{\frac{5}{4}}}$$

$$\mathcal{J}_{ij} = \overline{S}_{ik}\overline{S}_{kj} + \overline{\Omega}_{ik}\overline{\Omega}_{kj} - \frac{1}{3}\delta_{ij}(\overline{S}_{mn}\overline{S}_{mn} - \overline{\Omega}_{mn}\overline{\Omega}_{mn})$$

Effective relaxation time (see previous slide): $\tau_L^\star = \dfrac{(\nu+\nu_t) + \Delta t c_s^2/2}{c_s^2}$

Forced homogeneous isotropic turbulence

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

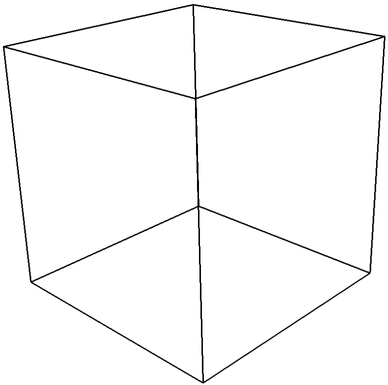
Forcing:

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

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

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

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.

Forced homogeneous isotropic turbulence

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

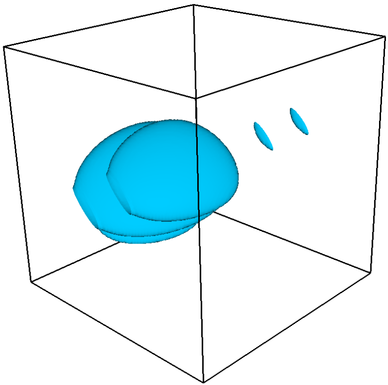
Forcing:

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

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

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

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



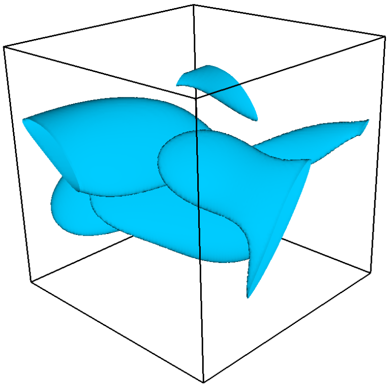
with phase

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

Forced 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.

Forced homogeneous isotropic turbulence

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

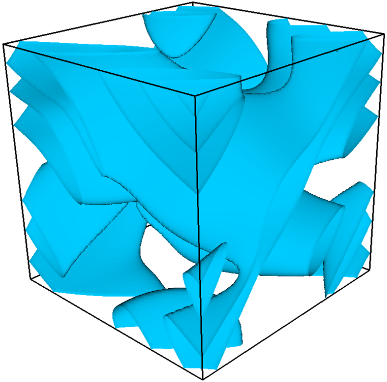
Forcing:

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

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

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

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



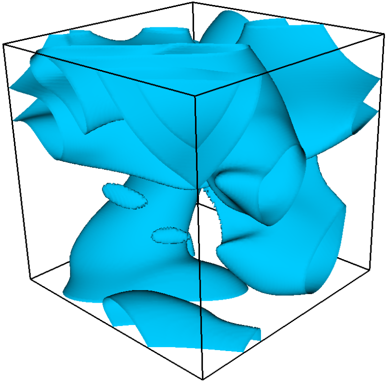
with phase

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

Forced 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.

Forced homogeneous isotropic turbulence

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

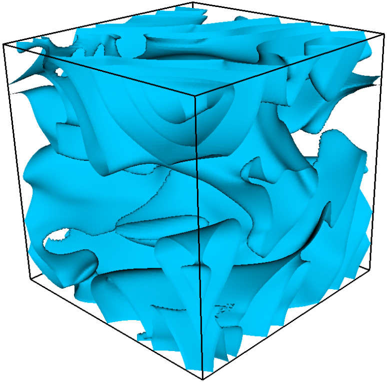
Forcing:

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

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

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

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.

Forced homogeneous isotropic turbulence

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

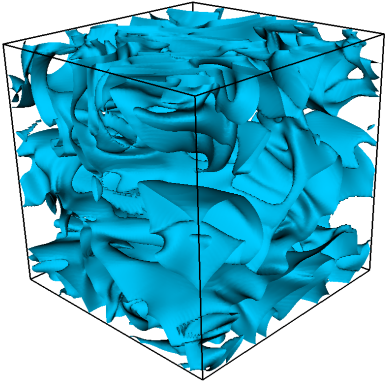
Forcing:

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

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

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

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.

Forced homogeneous isotropic turbulence

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

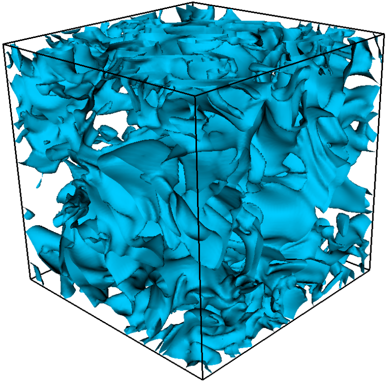
Forcing:

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

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

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

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



with phase

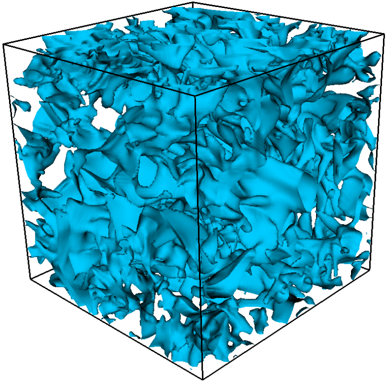
$$G(\kappa_x,\kappa_y,\kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right) \text{ for } (0 < \kappa_i \leq 2) \text{ and } \phi$$

being a random phase value.

Forced 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:

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

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

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

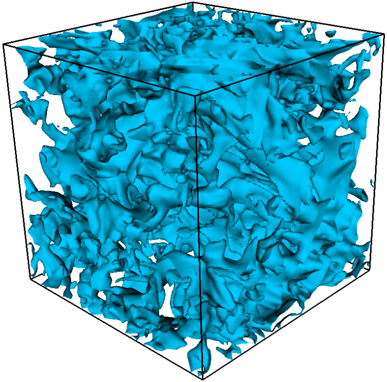
with phase

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

Forced 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:

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

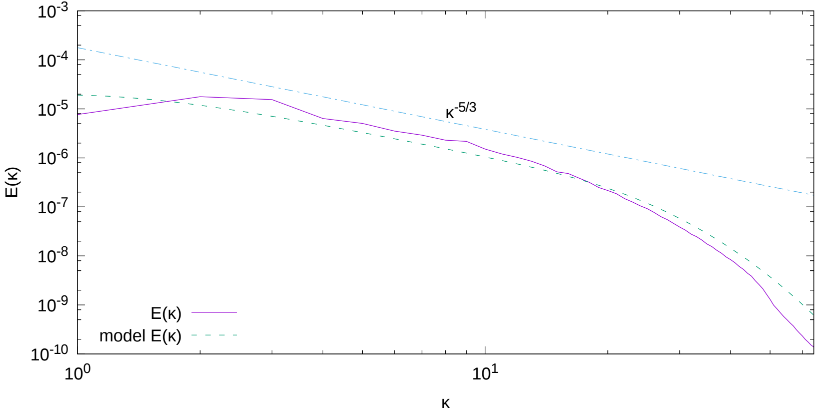
$$F_y = -A\left(\frac{\kappa_x\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z)$$

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

with phase

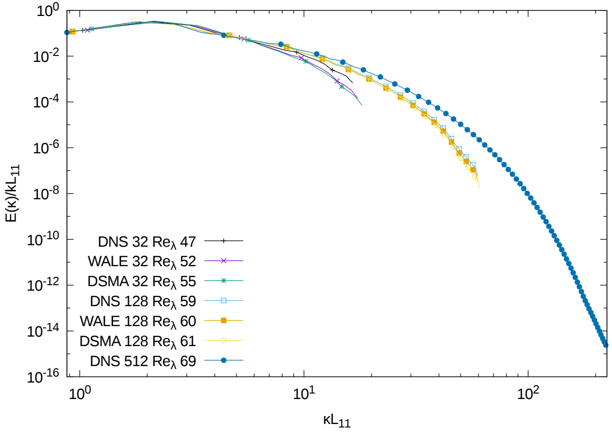
$G(\kappa_x,\kappa_y,\kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right)$ for $(0 < \kappa_i \le 2)$ and ϕ being a random phase value.

Comparison with model spectrum



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

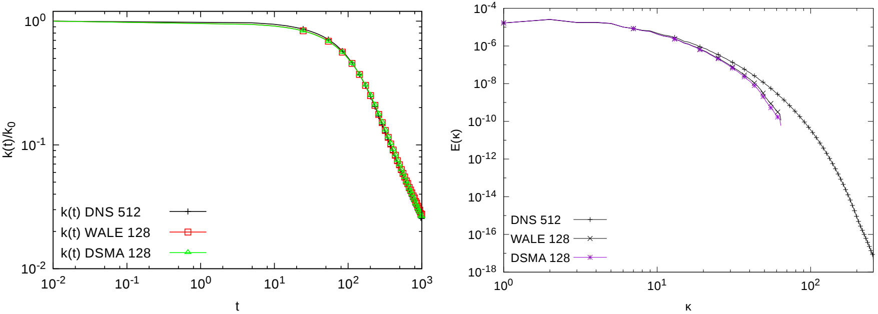
LES model spectra



Time-averaged energy spectra normalised by the turbulent kinetic energy k and the integral length scale L_{11} of LBM DNS and LES for two resolutions and DNS of the highest resolution for the viscosity value $\nu = 5 \cdot 10^{-5}$

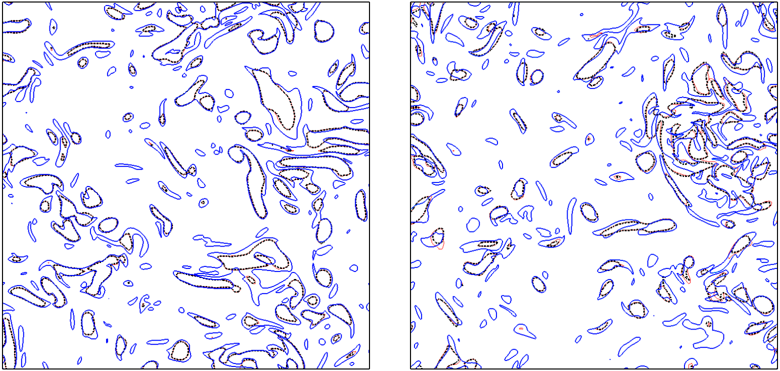
Decaying homogeneous isotropic turbulence

- Restart DNS of 512^3 resolution without forcing. Volume-averaging to 128^3 cells gives DSMA and WALE initial conditions



Evolution of the turbulent kinetic energy k (left) and energy spectra at $t = 68.72$ (right) for DNS of 512^3 against DSMA and WALE of 128^3 cells resolution.

Flow field comparison

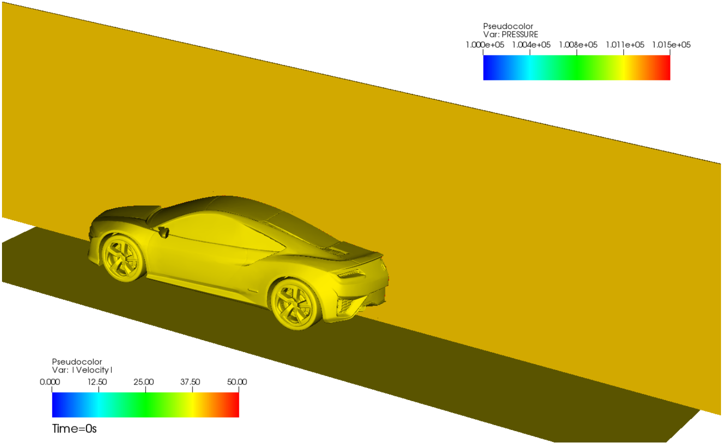


Contours of vorticity magnitude ($|\omega| = 0.18$) at $t = 4.91$ (left) and $t = 68.72$ (right) for DNS (thin blue lines) of 512^3 against DSMA (dotted black lines) and WALE (thick red lines) of 128^3 cells resolution

Outline

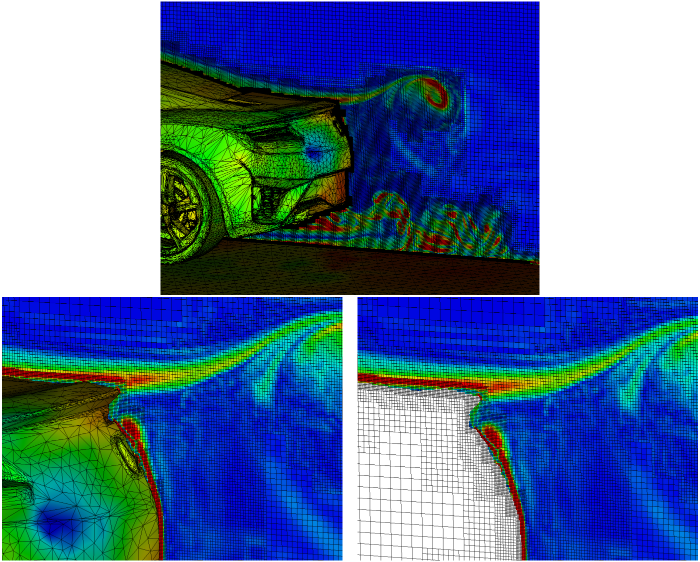
Wind tunnel simulation of a prototype car

Fluid velocity and pressure on vehicle



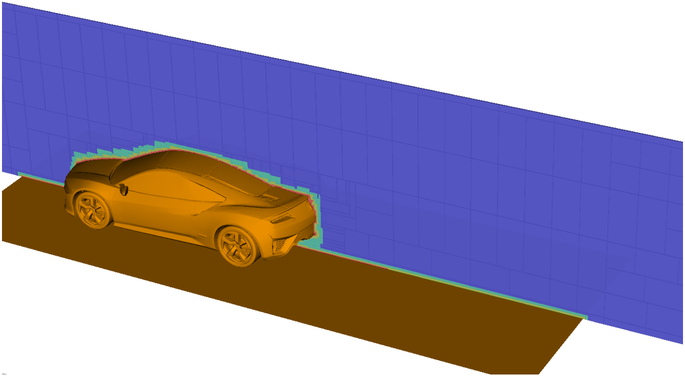
- Inflow 40 $\mathrm{m/s}$. LES model active. Characteristic boundary conditions.
- To $t = 0.5\,\mathrm{s}$ ($\sim$ 4 characteristic lengths) with 31,416 time steps on finest level in $\sim$ 37 h on 200 cores (7389 h CPU). Channel: 15 $\mathrm{m} \times$ 5 $\mathrm{m} \times$ 3.3 m

Mesh adaptation



Mesh adaptation

Used refinement blocks and levels (indicated by color)



- SAMR base grid 600 × 200 × 132 cells, $r_{1,2,3} = 2$ yielding finest resolution of $\Delta x = 3.125\,\mathrm{mm}$
- Adaptation based on level set and scaled gradient of magnitude of vorticity vector
- 236M cells vs. 8.1 billion (uniform) at $t = 0.4075\,\mathrm{s}$

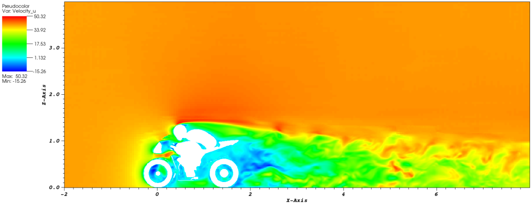
Refinement at $t = 0.4075\,\mathrm{s}$

Level	Grids	Cells
0	11,605	15,840,000
1	11,513	23,646,984
2	31,382	144,447,872
3	21,221	52,388,336

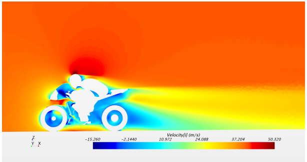
Flow over a motorcycle

- Inflow 40 $\mathrm{m/s}$. Bouzidi pressure boundary conditions at outflows. CSMA LES model active.
- SAMR base grid $200 \times 80 \times 80$ cells, $r_{1,2,3} = 2$ yielding finest resolution of $\Delta x = 6.25\,\mathrm{mm}$. 23560 time steps on finest level
- Forces in AMROC-LBM are time-averaged over interval $[0.5\mathrm{s}, 1\mathrm{s}]$
- Unstructured STAR-CCM+ mesh has significantly finer as well as coarser cells

AMROC-LBM LES at $t = 1\,\mathrm{s}$



STAR-CCM+ steady RANS



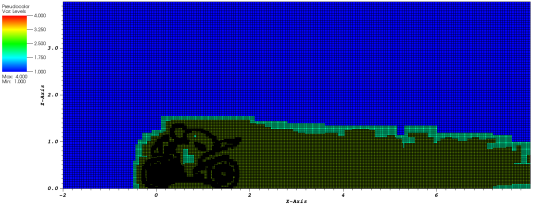
Velocity in flow direction

Variables	Forces (N) Drag	Sideforce	Lift	Total	Cores	Wall Time h	CPU Time h
STAR-CCM+	297	5	9	297	10	4.9	78
AMROC	297	10	23	298	64	10	635

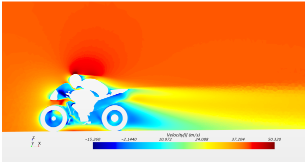
Flow over a motorcycle

- Inflow 40 $\mathrm{m/s}$. Bouzidi pressure boundary conditions at outflows. CSMA LES model active.
- SAMR base grid $200 \times 80 \times 80$ cells, $r_{1,2,3} = 2$ yielding finest resolution of $\Delta x = 6.25\,\mathrm{mm}$. 23560 time steps on finest level
- Forces in AMROC-LBM are time-averaged over interval $[0.5\mathrm{s}, 1\mathrm{s}]$
- Unstructured STAR-CCM+ mesh has significantly finer as well as coarser cells

AMROC-LBM LES at $t = 1\,\mathrm{s}$



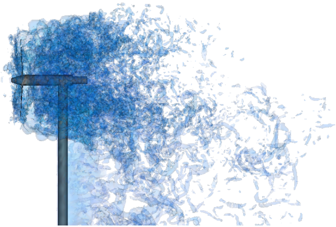
STAR-CCM+ steady RANS

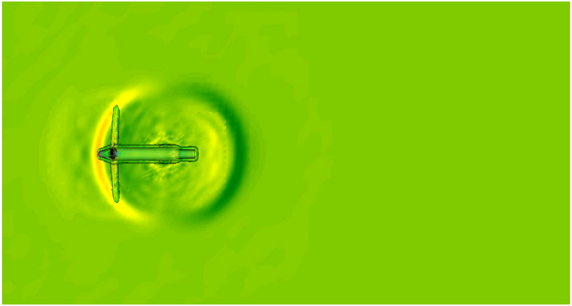


Velocity in flow direction

Variables	Forces (N) Drag	Sideforce	Lift	Total	Cores	Wall Time h	CPU Time h
STAR-CCM+	297	5	9	297	10	4.9	78
AMROC	297	10	23	298	64	10	635

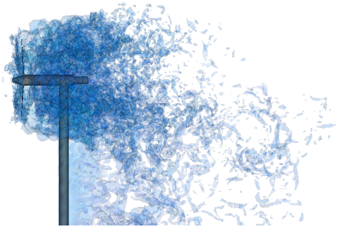
Mexico experimental turbine – 0^o inflow

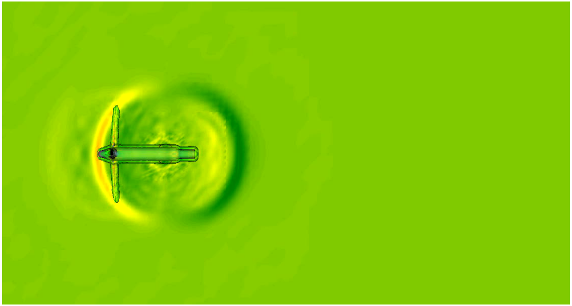




- Setup and measurements by Energy Research Centre of the Netherlands (ECN) and the Technical University of Denmark (DTU) [Schepers and Boorsma, 2012]
- Inflow velocity 14.93 $\mathrm{m/s}$ in wind tunnel of 9.5 $\mathrm{m} \times$ 9.5 m cross section.
- Rotor diameter $D =$ 4.5 m. Prescribed motion with 424.5 rpm: tip speed 100 $\mathrm{m/s}$, $\mathrm{Re}_r \approx$ 75839 TSR 6.70
- Simulation with three additional levels with factors 2, 2, 4. Resolution of rotor and tower $\Delta x =$ 1.6 cm
- 149.5 h on 120 cores Intel-Xeon (17490 h CPU) for 10 s

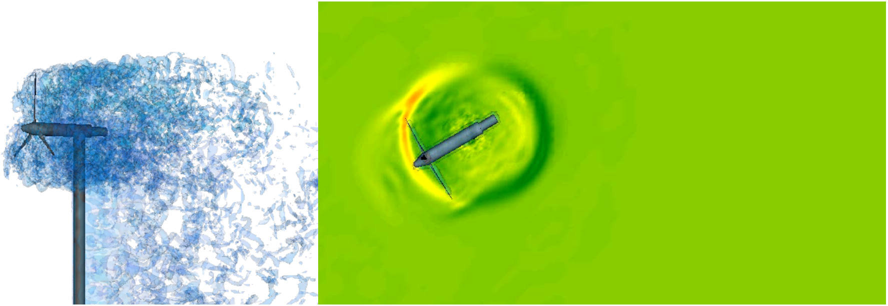
Mexico experimental turbine – 0^o inflow





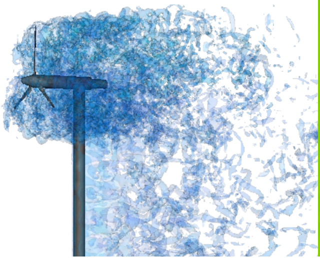
- Setup and measurements by Energy Research Centre of the Netherlands (ECN) and the Technical University of Denmark (DTU) [Schepers and Boorsma, 2012]
- Inflow velocity 14.93 $\mathrm{m/s}$ in wind tunnel of 9.5 $\mathrm{m} \times$ 9.5 m cross section.
- Rotor diameter $D = 4.5\,\mathrm{m}$. Prescribed motion with 424.5 rpm: tip speed 100 $\mathrm{m/s}$, $\mathrm{Re}_r \approx 75839$ TSR 6.70
- Simulation with three additional levels with factors 2, 2, 4. Resolution of rotor and tower $\Delta x = 1.6\,\mathrm{cm}$
- 149.5 h on 120 cores Intel-Xeon (17490 h CPU) for 10 s
- Data collected as average during $t \in [5, 10]$. Load on blade 1 as it passes through $\theta = 0^o$ (pointing vertically upwards), 35 rotations

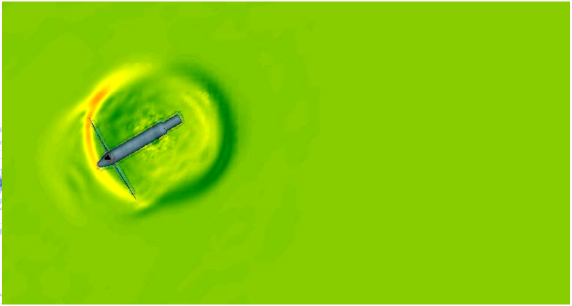
Mexico experimental turbine – 30^o yaw



- 157.6 h on 120 cores Intel-Xeon for 10 s (70.75 revolutions) $\longrightarrow \sim 22.25\,\mathrm{h}$ CPU/1M cells/revolution
- $\sim 12\,\mathrm{M}$ cells in total – level 0: 768,000, level 1: $\sim 1.5\,\mathrm{M}$, level 2: $\sim 6.8\,\mathrm{M}$, level 3: $\sim 3.0\,\mathrm{M}$

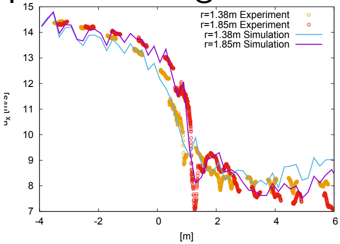
Mexico experimental turbine – 30^o yaw



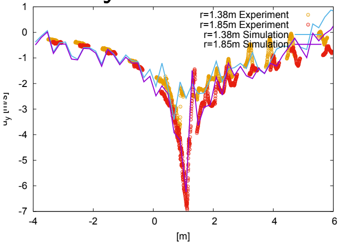


- 157.6 h on 120 cores Intel-Xeon for 10 s (70.75 revolutions) $\longrightarrow \sim$ 22.25 h CPU/1M cells/revolution
- $\sim$ 12 M cells in total – level 0: 768,000, level 1: $\sim$ 1.5 M, level 2: $\sim$ 6.8 M, level 3: $\sim$ 3.0 M
- For comparison [Schepers and Boorsma, 2012]:
- Wind Multi-Block Liverpool University (34 M cells): 209 h CPU/1M cells/revolution
- EllipSys3D (28.3 M cell mesh): $\sim$ 40.7 h CPU/1M cells/revolution, but $\sim$ 15% error in F_x and T_x already for 0° inflow [Sørensen et al., 2014]

Comparison along transects – 30^o yaw



u_x

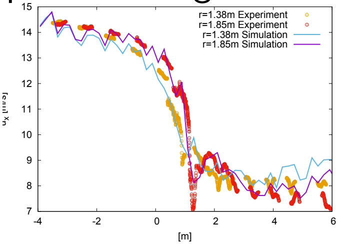


u_y

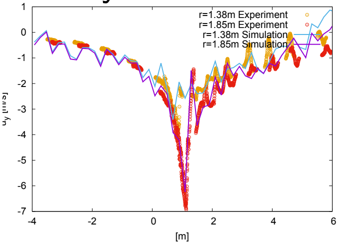
axial

RD, S. L. Wood. Proc. of TORQUE 2016. *J. Phys. Conference Series* 753: 082005, 2016.

Comparison along transects – 30^o yaw

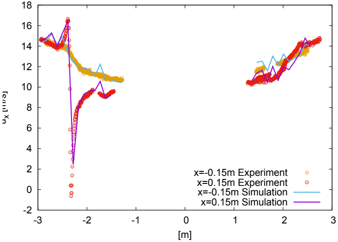


u_x

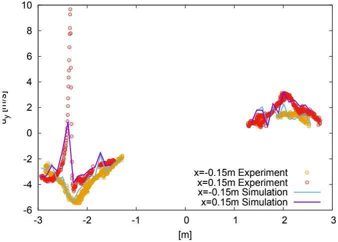


u_y

axial



u_x

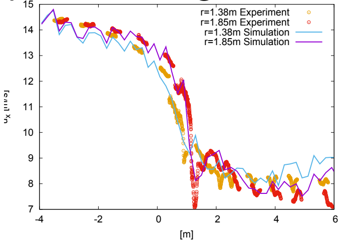


u_y

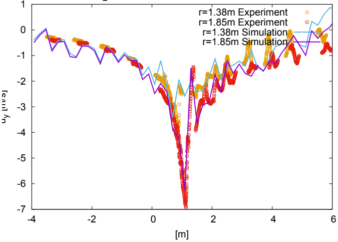
radial

RD, S. L. Wood. Proc. of TORQUE 2016. *J. Phys. Conference Series* 753: 082005, 2016.

Comparison along transects – 30^o yaw

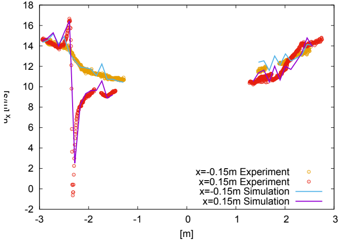


u_x

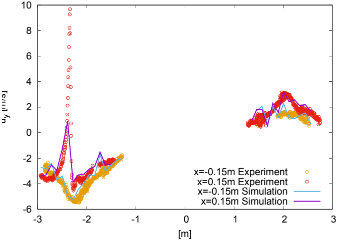


u_y

axial



u_x



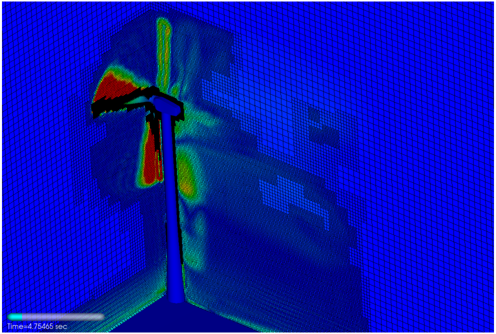
u_y

radial

- Blade loads: F_x: $\mathrm{Ref} = 13.66\,\mathrm{N}$, $\mathrm{cur.} = 14.8\,\mathrm{N}$ (8.3%)
- T_x: $\mathrm{Ref} = 7.72\,\mathrm{Nm}$, $\mathrm{cur.} = 8.36\,\mathrm{Nm}$ (8.3%)

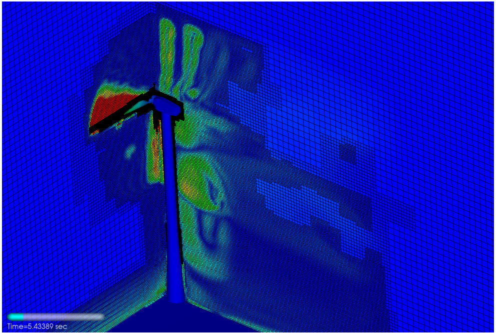
RD, S. L. Wood. Proc. of TORQUE 2016. *J. Phys. Conference Series* 753: 082005, 2016.

Single Vestas V27



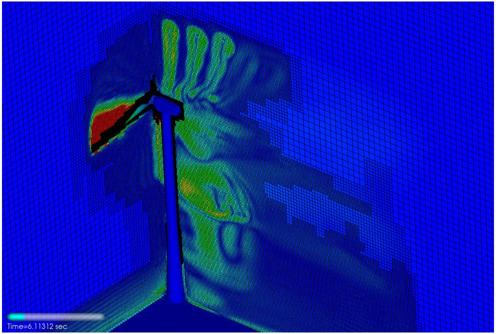
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

Single Vestas V27



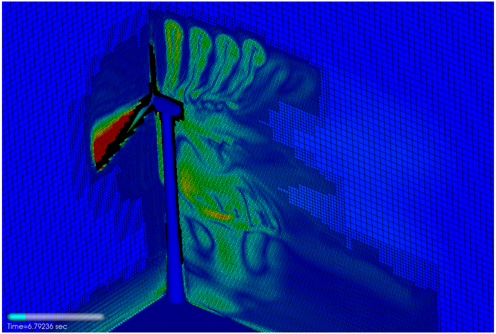
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- $\sim$ 24 time steps for 1^o rotation

Single Vestas V27



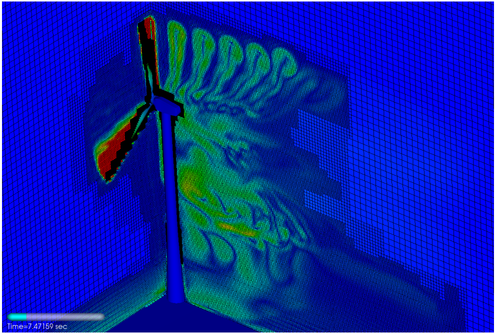
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_\mathrm{rpm} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

Single Vestas V27



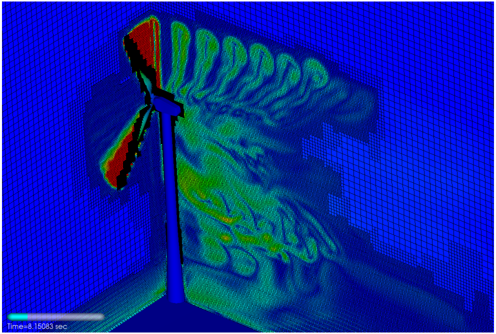
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

Single Vestas V27



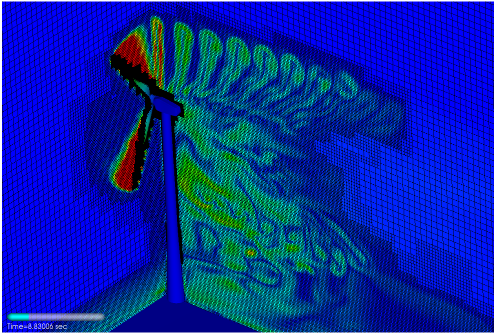
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

Single Vestas V27



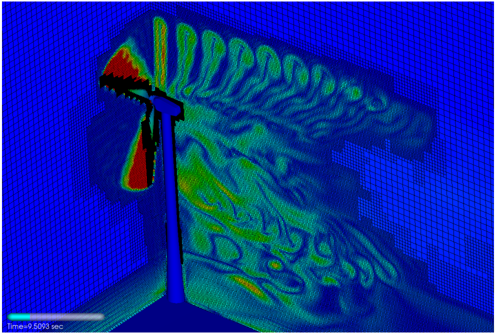
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

Single Vestas V27



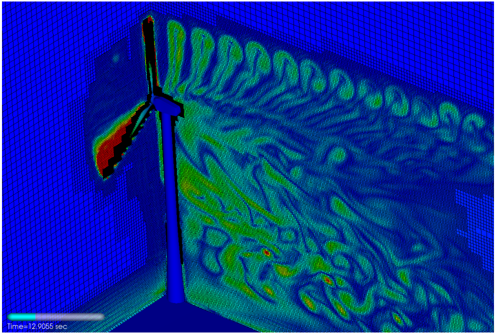
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- $\sim$ 24 time steps for 1^o rotation

Single Vestas V27



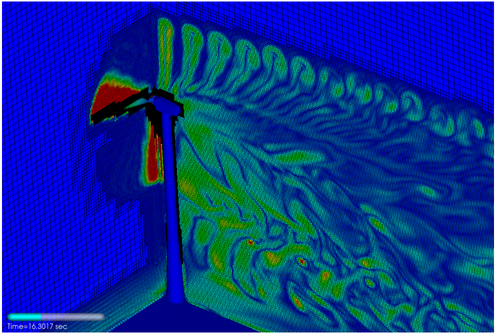
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors 2, 2, 4.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

Single Vestas V27



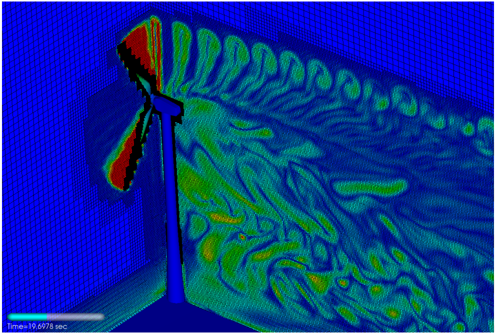
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

Single Vestas V27



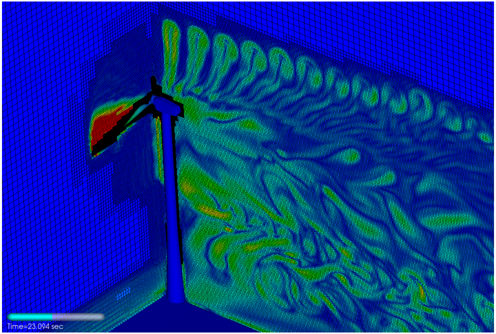
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

Single Vestas V27



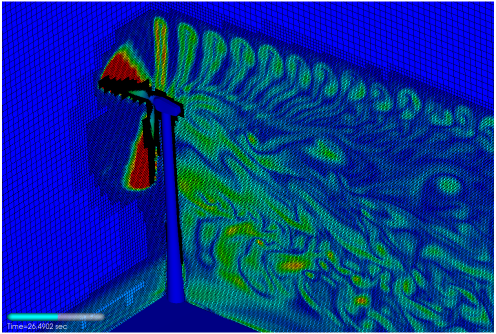
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

Single Vestas V27



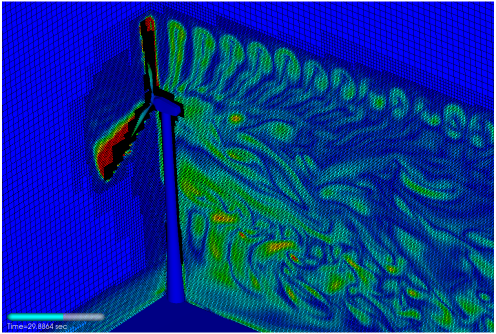
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

Single Vestas V27



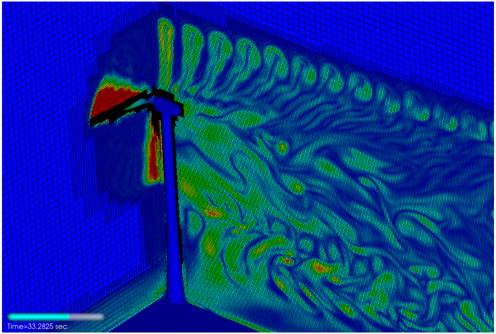
- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_\mathrm{rpm} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors 2, 2, 4.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

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.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

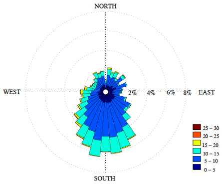
Single Vestas V27

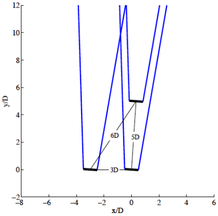


- Inflow velocity $u_\infty = 8\,\mathrm{m/s}$. Prescribed motion of rotor with $n_{\mathrm{rpm}} = 33$, $r = 14.5\,\mathrm{m}$: tip speed $46.7\,\mathrm{m/s}$, $\mathrm{Re}_r \approx 919,700$ TSR 5.84
- Simulation with three additional levels with refinement factors $2, 2, 4$.
- Refinement based on vorticity and level set.
- Sampled rotor and circular regions ($r_c = 1.5r$) every $0.034\,\mathrm{s}$ over $t = [8, 18]\,\mathrm{s}$
- Computing 84,806 highest level iterations to $t_e = 18\,\mathrm{s}$.
- ~ 24 time steps for 1^o rotation

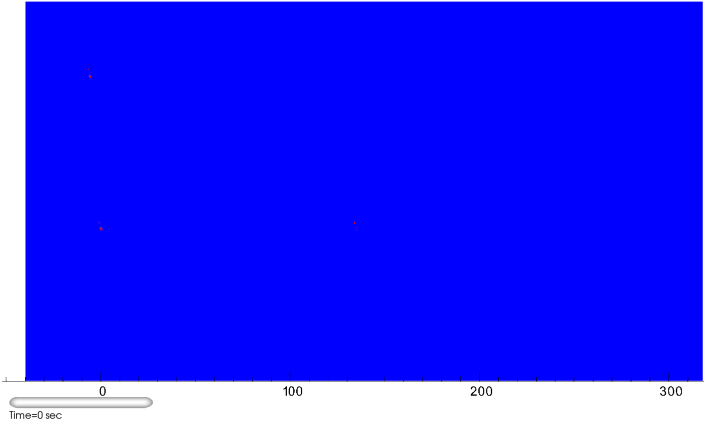
Simulation of the SWIFT array

- Three Vestas V27 turbines (geometric details prototypical). $225\,\mathrm{kW}$ power generation at wind speeds 14 to $25\,\mathrm{m/s}$ (then cut-off)
- Prescribed motion of rotor with 33 and $43\,\mathrm{rpm}$. 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, 2016]



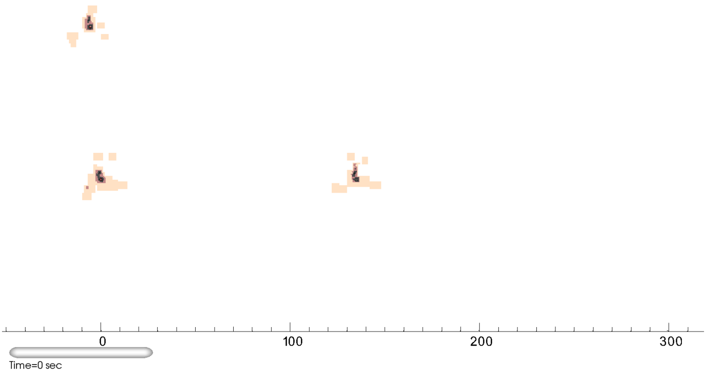


Vorticity – inflow at 30^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



- 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)
- $\sim 6.01\,\mathrm{h}$ per revolution ($961\,\mathrm{h}$ CPU) $\longrightarrow \sim 5.74\,\mathrm{h}$ CPU/1M cells/revolution

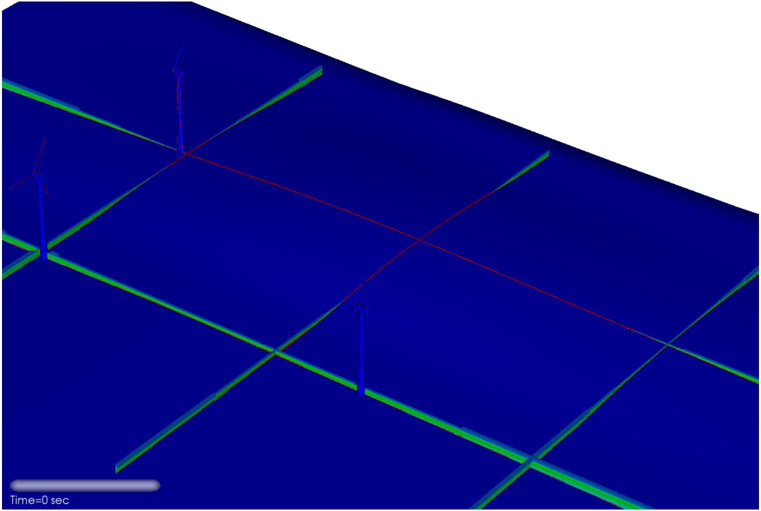
Levels – inflow at 30°, $u = 8\,\mathrm{m/s}$, 33 rpm



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

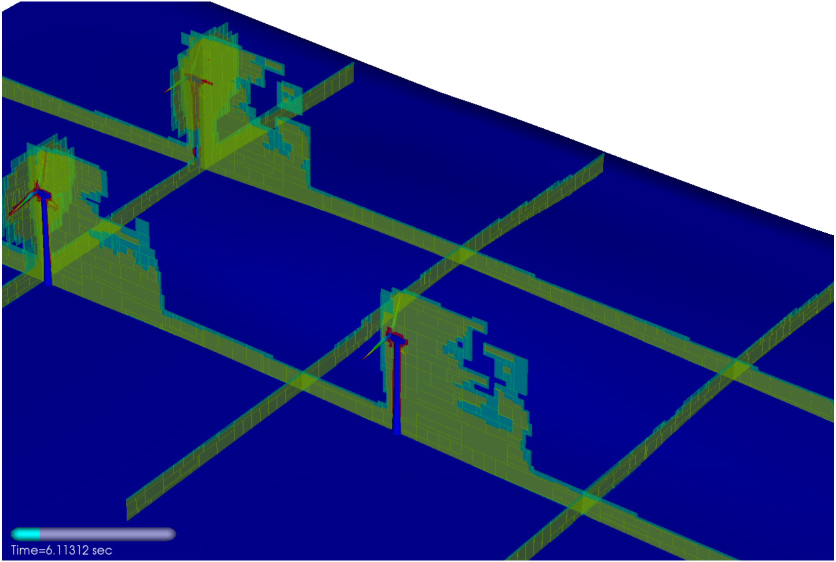
Level	Grids	Cells
0	2,463	10,752,000
1	6,464	20,674,760
2	39,473	131,018,832
3	827	4,909,632

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

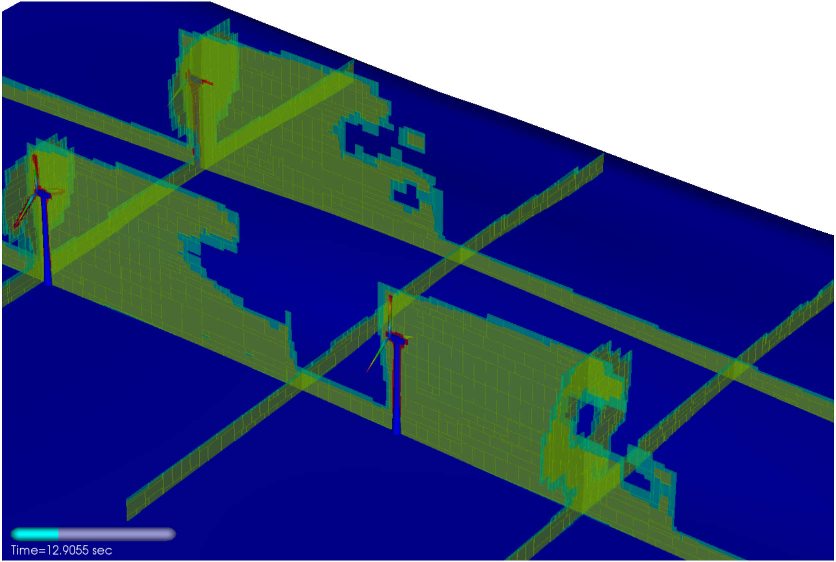


- Refinement of wake up to level 2 ($\Delta x = 25\,\mathrm{cm}$).
- Vortex break-up before 2nd turbine is reached.

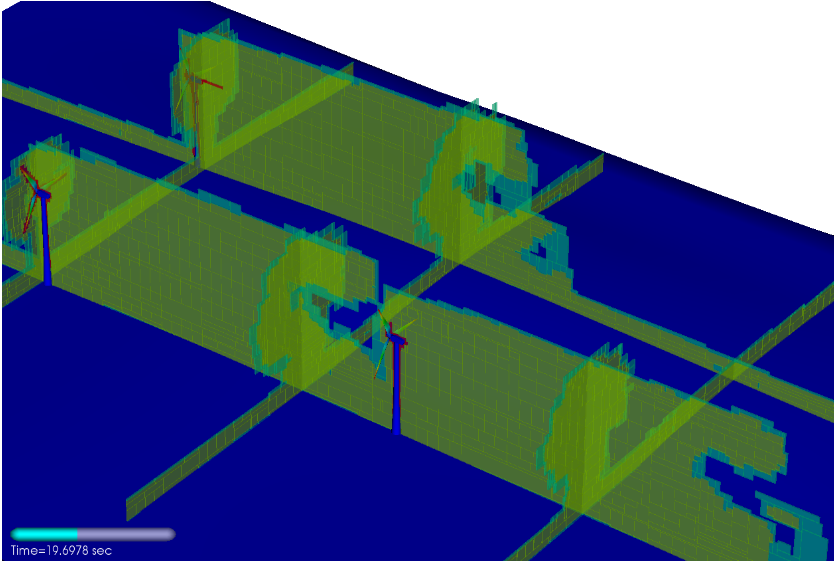
Refinement – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



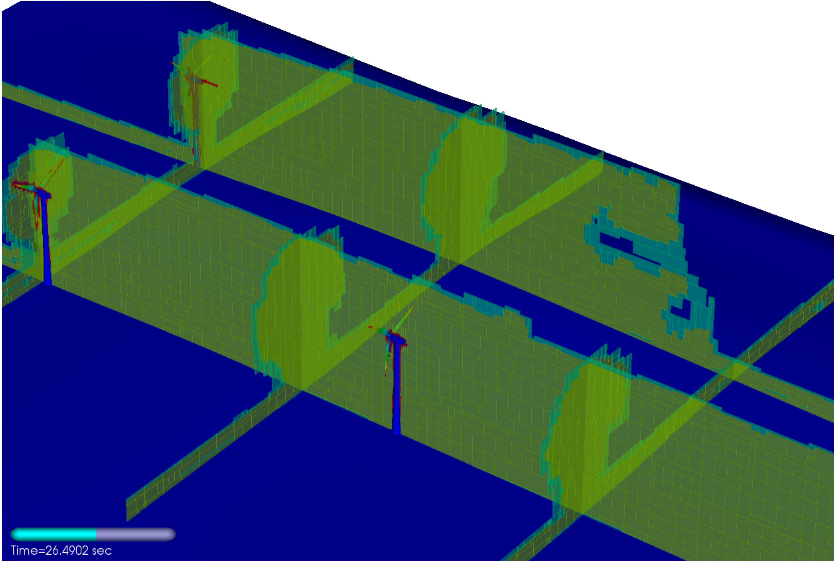
Refinement – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



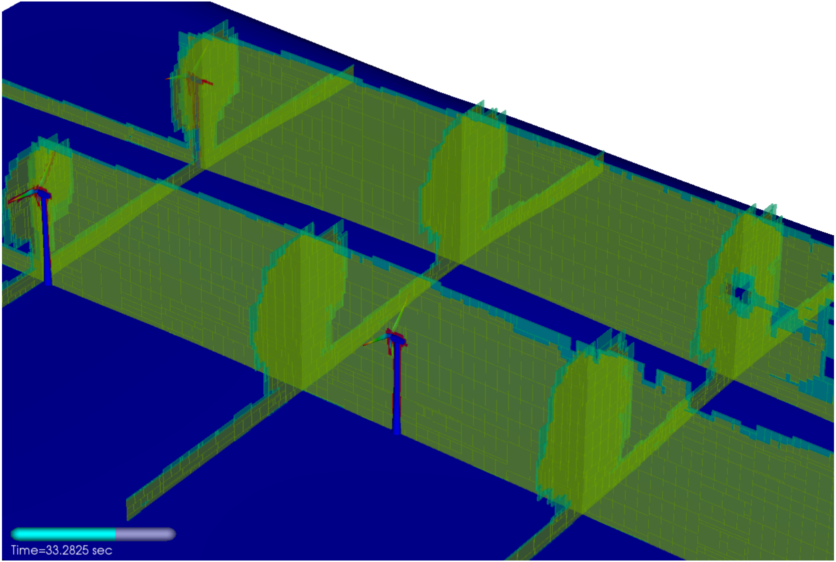
Refinement – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



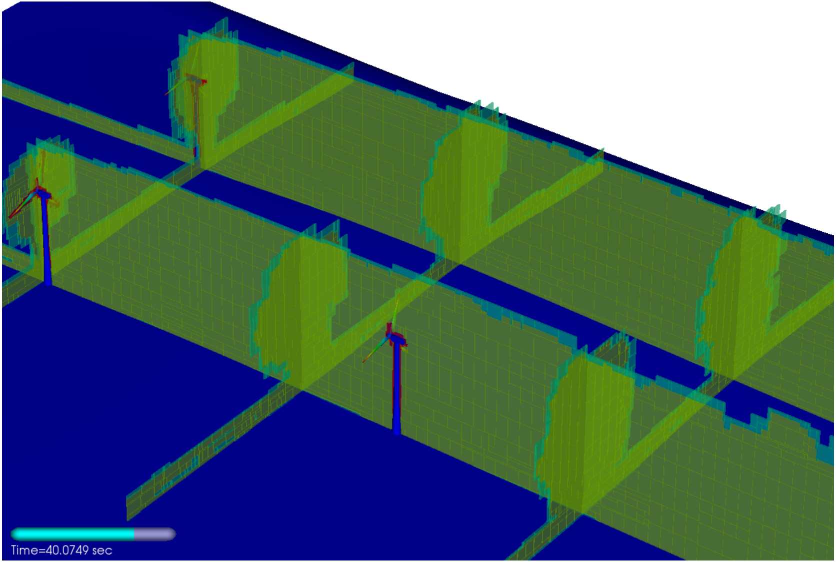
Refinement – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



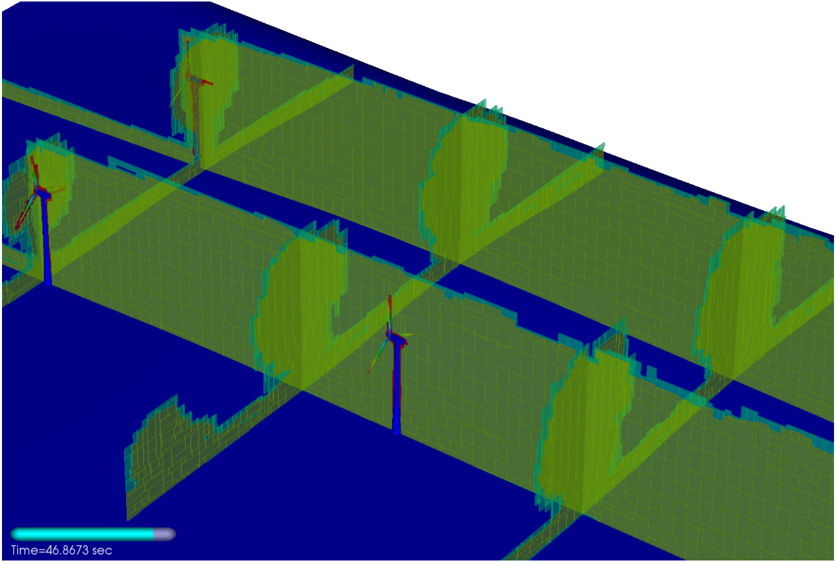
Refinement – inflow at 0^o, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$



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

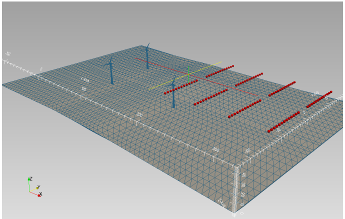


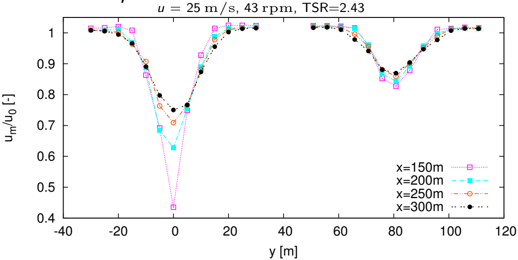
Refinement – inflow at 0°, $u = 8\,\mathrm{m/s}$, $33\,\mathrm{rpm}$

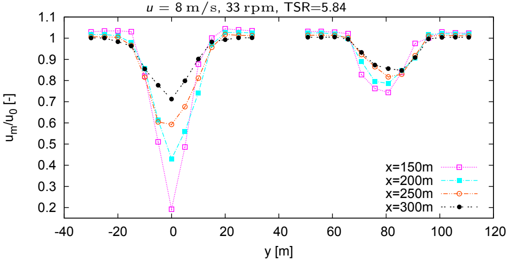


Mean point values – inflow at 0^o,

- 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





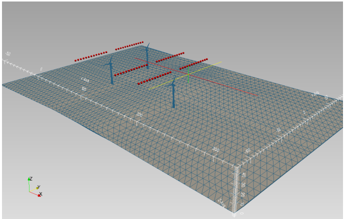


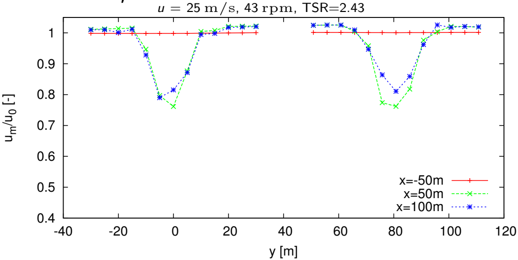
- Velocity deficits larger for higher TSR.

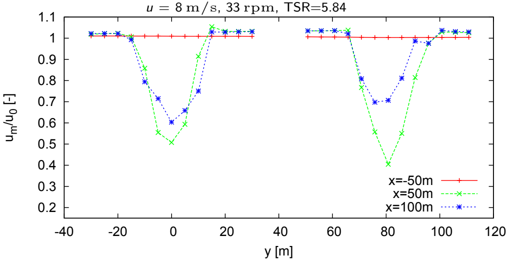
RD, S. L. Wood. New Results in Numerical and Experimental Fluid Mechanics X, pages 845-857, Springer, 2016.

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.

RD, S. L. Wood. New Results in Numerical and Experimental Fluid Mechanics X, pages 845-857, Springer, 2016.

Lattice Boltzmann equation in mapped coordinates

Consider mapping from Cartesian to non-Cartesian coordinates

$$\xi = \xi(x,y),\ \eta = \eta(x,y)$$

with

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial}{\partial \eta}\frac{\partial \eta}{\partial x},\ \frac{\partial}{\partial y} = \frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial}{\partial \eta}\frac{\partial \eta}{\partial y}.$$

Under this transformation the convection term reads

$$\begin{aligned}
\mathbf{e}_\alpha \cdot \nabla f_\alpha &= e_{\alpha x}\frac{\partial f_\alpha}{\partial x} + e_{\alpha y}\frac{\partial f_\alpha}{\partial y} \\
&= e_{\alpha x}\left(\frac{\partial f_\alpha}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial f_\alpha}{\partial \eta}\frac{\partial \eta}{\partial x}\right) + e_{\alpha y}\left(\frac{\partial f_\alpha}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial f_\alpha}{\partial \eta}\frac{\partial \eta}{\partial y}\right) \\
&= \left(e_{\alpha x}\frac{\partial \xi}{\partial x} + e_{\alpha y}\frac{\partial \xi}{\partial y}\right)\frac{\partial f_\alpha}{\partial \xi} + \left(e_{\alpha x}\frac{\partial \eta}{\partial x} + e_{\alpha y}\frac{\partial \eta}{\partial y}\right)\frac{\partial f_\alpha}{\partial \eta} \\
&= \tilde{e}_{\alpha\xi}\frac{\partial f_\alpha}{\partial \xi} + \tilde{e}_{\alpha\eta}\frac{\partial f_\alpha}{\partial \eta},
\end{aligned}$$

and hence the lattice Boltzmann equation becomes

$$\frac{\partial f}{\partial t} + \tilde{e}_{\alpha\xi}\frac{\partial f_\alpha}{\partial \xi} + \tilde{e}_{\alpha\eta}\frac{\partial f_\alpha}{\partial \eta} = -\frac{1}{\tau}\left(f_\alpha - f_\alpha^{eq}\right).$$

Scheme construction

Currently using the explicit 4th-order Runge-Kutta scheme

$$f_\alpha^1 = f_\alpha^t,\ f_\alpha^2 = f_\alpha^1 + \frac{\Delta t}{4} R_\alpha^1,$$
$$f_\alpha^3 = f_\alpha^1 + \frac{\Delta t}{3} R_\alpha^2, f_\alpha^4 = f_\alpha^1 + \frac{\Delta t}{2} R_\alpha^3,$$
$$f_\alpha^{t+\Delta t} = f_\alpha^1 + \Delta t R_\alpha^4.$$

with

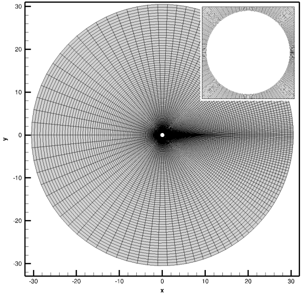
$$R_{\alpha(i,j)} = -\left(\tilde{e}_{\alpha\xi_{(i,j)}} \frac{f_{\alpha(i+1,j)} - f_{\alpha(i-1,j)}}{2\Delta\xi} + \tilde{e}_{\alpha\eta_{(i,j)}} \frac{f_{\alpha(i,j+1)} - f_{\alpha(i,j-1)}}{2\Delta\eta} \right) - \frac{1}{\tau} \left(f_{\alpha(i,j)} - f^{eq}_{\alpha(i,j)} \right)$$

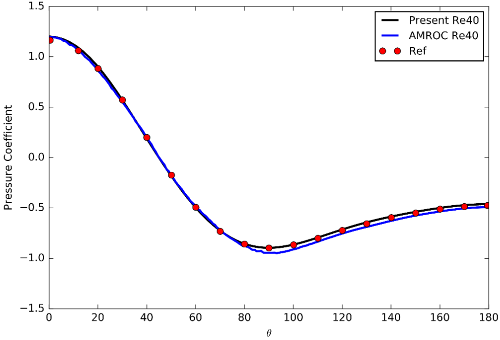
for the solution, 2nd-order central differences to approximate derivatives.
A 4th-order dissipation term

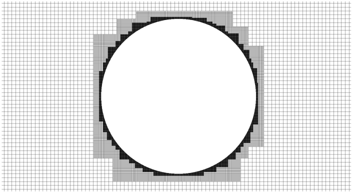
$$D = -\epsilon \left((\Delta\xi)^4 \frac{\partial^4 f_\alpha}{\partial \xi^4} + (\Delta\eta)^4 \frac{\partial^4 f_\alpha}{\partial \eta^4} \right)$$

is added for stabilization [Hejranfar and Hajihassanpour, 2017].
Prototype implementation is presently on finite difference meshes!

2d cylinder study



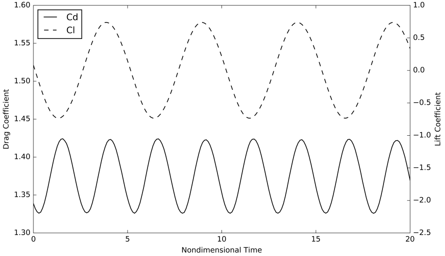




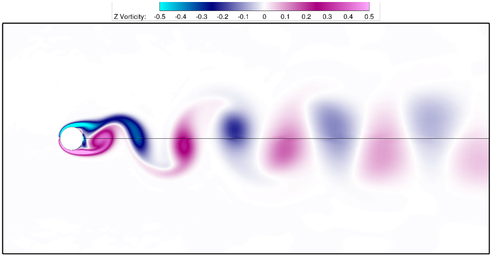
Re	Author(s)	C_d	$C_p(0)$	$C_p(180)$	$2L/D$
20	[Tritton, 1959]	2.20	-	-	-
	[Henderson, 1995]	2.06	-	-0.60	-
	[Dennis and Chang, 1970]	2.05	1.27	-0.58	1.88
	[Hejranfar and Ezzatneshan, 2014]	2.02	1.25	-0.59	1.84
	AMROC-LBM	1.98	1.26	-0.59	1.85
	Present	2.02	1.31	-0.55	1.85
40	[Tritton, 1959]	1.65	-	-	-
	[Henderson, 1995]	1.55	-	-0.53	-
	[Dennis and Chang, 1970]	1.52	1.14	-0.50	4.69
	[Hejranfar and Ezzatneshan, 2014]	1.51	1.15	-0.48	4.51
	AMROC-LBM	1.45	1.19	-0.49	4.66
	Present	1.51	1.19	-0.46	4.60

$2L/D$ is normalized length of wake behind cylinder

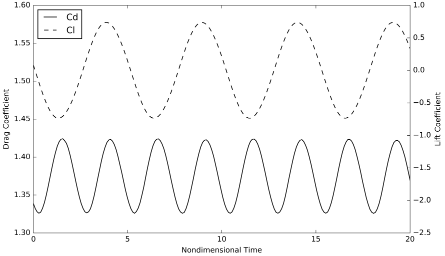
2d cylinder study – unsteady flow case



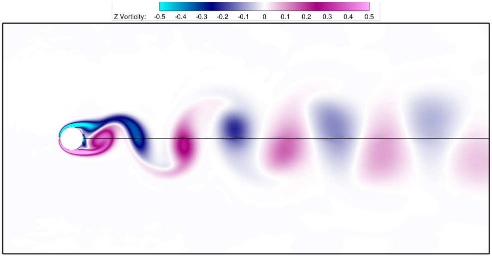
Re	Author(s)	St	$\overline{C_d}$	C_l'
100	[Chiu et al., 2010]	0.167	1.35	0.30
	AMROC-LBM	0.166	1.28	0.32
	Present	0.165	1.36	0.35
200	[Chiu et al., 2010]	0.198	1.37	0.71
	AMROC-LBM	0.196	1.26	0.70
	Present	0.196	1.37	0.73



2d cylinder study – unsteady flow case



Re	Author(s)	St	$\overline{C_d}$	C_l'
100	[Chiu et al., 2010]	0.167	1.35	0.30
	AMROC-LBM	0.166	1.28	0.32
	Present	0.165	1.36	0.35
200	[Chiu et al., 2010]	0.198	1.37	0.71
	AMROC-LBM	0.196	1.26	0.70
	Present	0.196	1.37	0.73



Re		CPU-time	Mesh
20	AMROC-LBM	24:55:21	297796
	Present	06:08:41	65536
40	AMROC-LBM	27:10:08	317732
	Present	05:57:17	65536
100	AMROC-LBM	113:15:37	1026116
	Present	05:58:49	65536
200	AMROC-LBM	130:37:18	1130212
	Present	06:03:42	65536

Conclusions – subsonic aerodynamics with LBM

- Cartesian LBM is a very efficient low-dissipation method for subsonic aerodynamic simulation and especially suitable for DNS and LES
- Cartesian CFD with block-based AMR is faster than cell-cased AMR and tailored for modern massively parallel computer systems
- Fast dynamic mesh adaptation in AMROC makes FSI problems with complex motion easily accessible. Time-explicit approach leads to very tight coupling
- For high Reynolds number flows around complex bodies an LES turbulence model is vital for stability (so are higher-order in- and outflow boundary conditions)
- Currently validating and extending (dynamic) Smagorinsky with wall-near damping and WALE model for realistic problems
- Turbulent wall function boundary condition model under development

Conclusions – subsonic aerodynamics with LBM

- Cartesian LBM is a very efficient low-dissipation method for subsonic aerodynamic simulation and especially suitable for DNS and LES
- Cartesian CFD with block-based AMR is faster than cell-cased AMR and tailored for modern massively parallel computer systems
- Fast dynamic mesh adaptation in AMROC makes FSI problems with complex motion easily accessible. Time-explicit approach leads to very tight coupling
- For high Reynolds number flows around complex bodies an LES turbulence model is vital for stability (so are higher-order in- and outflow boundary conditions)
- Currently validating and extending (dynamic) Smagorinsky with wall-near damping and WALE model for realistic problems
- Turbulent wall function boundary condition model under development
- Accurate simulation of thin, wall-resolved boundary layers is dramatically more efficient with the non-Cartesian LBM approach, despite the availability of AMR in AMROC
    - Develop non-Cartesian version of AMROC-LBM as near-term goal
    - Chimera technique within AMROC-LBM might be long-term goal

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.

[Chiu et al., 2010] Chiu, P. H., Lin, R. K., and Sheu, T. W. (2010). A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier Stokes equations in time-varying complex geometries. *Journal of Computational Physics*.

[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 and Wood, 2016] Deiterding, R. and Wood, S. L. (2016). An adaptive lattice Boltzmann method for predicting wake fields behind wind turbines. In Dillmann, A., Heller, G., Krämer, E., Wagner, C., and Breitsamter, C., editors, *New Results in Numerical and Experimental Fluid Mechanics X*, volume 132 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 845–857. Springer.

[Dennis and Chang, 1970] Dennis, S. C. R. and Chang, G. (1970). Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. *Journal of Fluid Mechanics*, 42(03):471.

[Fusegi et al., 1991] Fusegi, T., Hyun, J., Kuwahara, K., and Farouk, B. (1991). A numerical study of three-dimensional natural convection in a differentially heated cubical enclosure. *Int. J. Heat and Mass Transfer*, 34:1543–1557.

[Guo et al., 2002] Guo, Z., Shi, B., and Zheng, C. (2002). A coupled lattice BGK model for the Boussinesq equations. *Int. J. Numerical Methods in Fluids*, 39:325–342.

[Hähnel, 2004] Hähnel, D. (2004). *Molekulare Gasdynamik*. Springer.

[Hejranfar and Ezzatneshan, 2014] Hejranfar, K. and Ezzatneshan, E. (2014). Implementation of a high-order compact finite-difference lattice Boltzmann method in generalized curvilinear coordinates. *Journal of Computational Physics*, 267:28–49.

[Hejranfar and Hajihassanpour, 2017] Hejranfar, K. and Hajihassanpour, M. (2017). Chebyshev collocation spectral lattice Boltzmann method in generalized curvilinear coordinates. *Computers and Fluids*.

References II

[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, 2003] Mauch, S. P. (2003). *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*. PhD thesis, California Institute of Technology.

[Nazarinia et al., 2012] Nazarinia, M., Jacono, D. L., Thompson, M. C., and Sheridan, J. (2012). Flow over a cylinder subjected to combined translational and rotational oscillations. *J. Fluids and Structures*, 32:135–145.

[Schepers and Boorsma, 2012] Schepers, J. G. and Boorsma, K. (2012). Final report of iea task 29: Mexnext (phase 1) – Analysis of Mexico wind tunnel measurements. Technical Report ECN-E-12-004, European research Centre of the Netherlands.

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

[Sørensen et al., 2014] Sørensen, N. N., Bechmann, A., Rethore, P. E., and Zahle, F. (2014). Near wake reynolds-averaged Navier-Stokes predictions of the wake behind the MEXICO rotor in axial and yawed flow conditions. *Wind Energy*, 17:75–86.

[Toomey and Eldredge, 2008] Toomey, J. and Eldredge, J. D. (2008). Numerical and experimental study of the fluid dynamics of a flapping wing with low order flexibility. *Physics of Fluids*, 20(7):073603.

[Tritton, 1959] Tritton, D. (1959). Experiments on the flow past a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics*, 6(4):547–567.

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

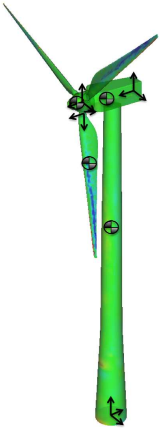
[Wood and Deiterding, 2015] Wood, S. L. and Deiterding, R. (2015). A lattice Boltzmann method for horizontal axis wind turbine simulation. In *14th Int. Conf. on Wind Engineering*.

[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]

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

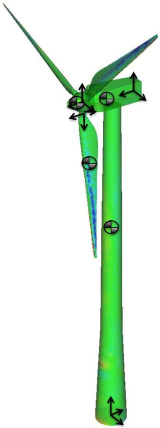


m = mass of the body, 1 = the 4×4 homogeneous identity matrix,
$\mathbf{a}_p$ = acceleration of link frame with origin at $\mathbf{p}$ in the preceding link's frame,
$\mathbf{I}_{\mathrm{cm}}$ = moment of inertia about the center of mass,
$\boldsymbol{\omega}$ = angular velocity of the body,
$\boldsymbol{\alpha}$ = 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.