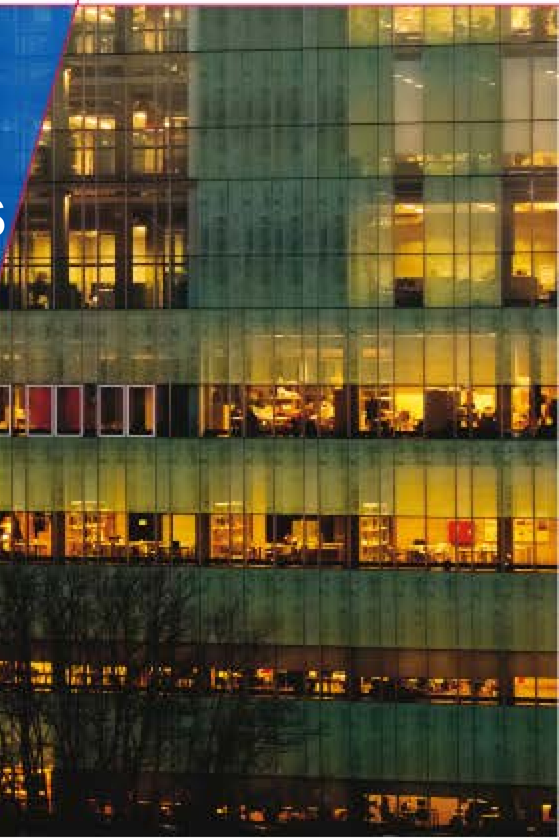
Excerpt from the Proceedings of the 2012 COMSOL Conference in Milan

3D Modeling of Urban Areas for Built Environment CFD Applications using Comsol

dr. Jos van Schijndel



TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

Scale levels Building physics

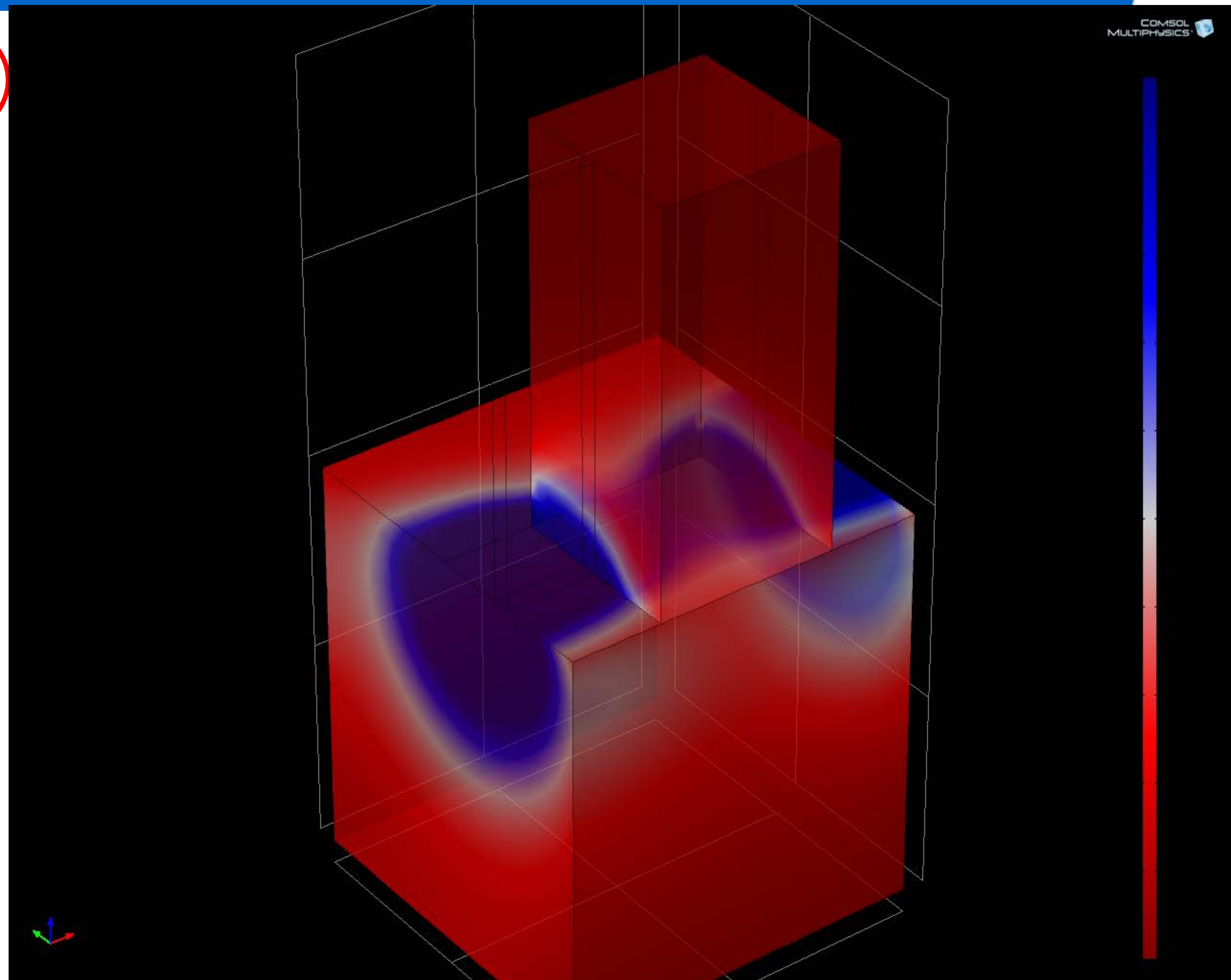
Scale levels, from left to right: EU; Urban area; Building; Material;

- [mm] Material Physics
- [m] Building Physics
- [km] Urban Physics
- [Mm] Climate Physics

Technische Universiteit Eindhoven University of Technology

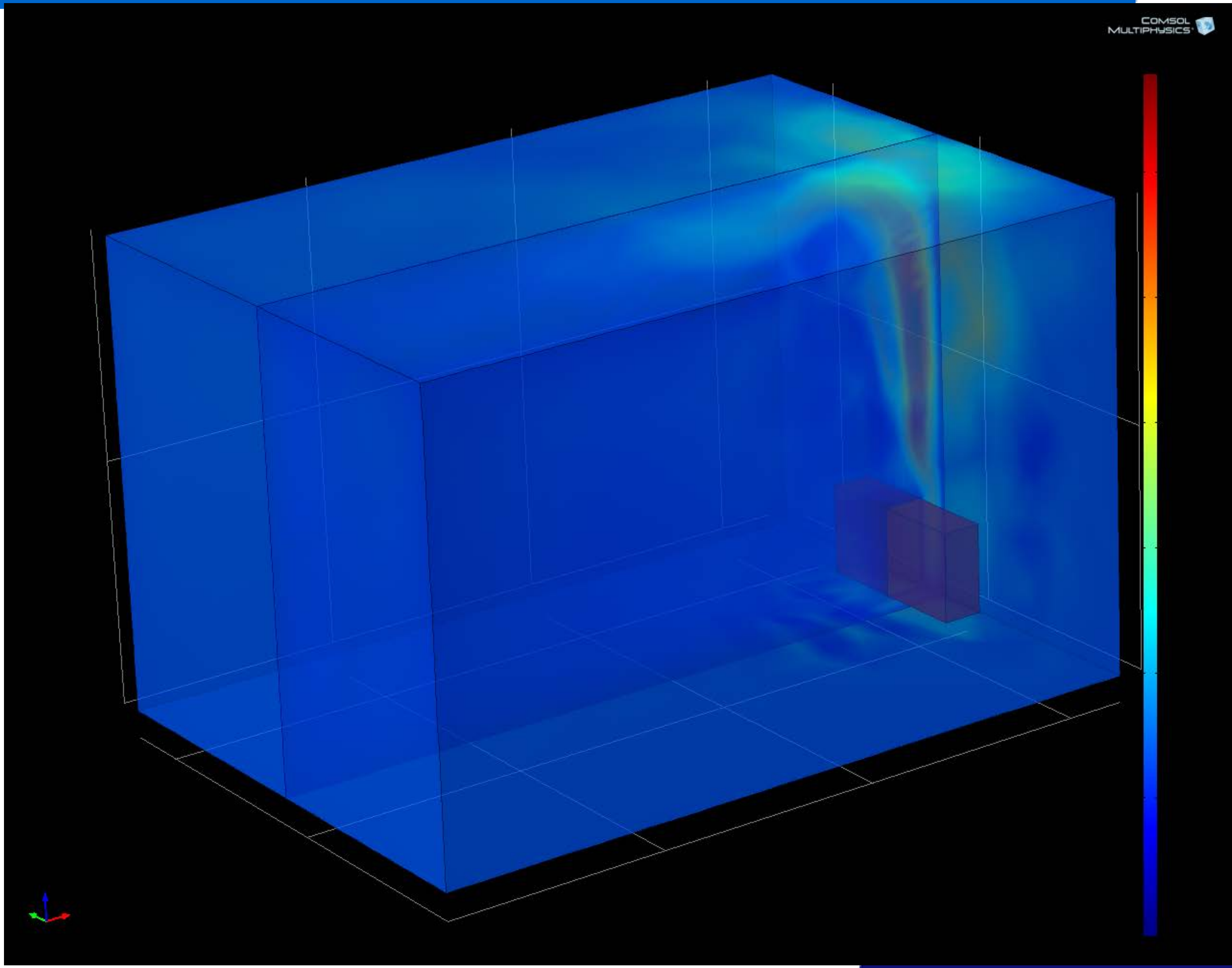
Scale level [mm] Material Physics
Moisture induced damages

Scale levels, from left to right: EU; Urban area; Building; Material;

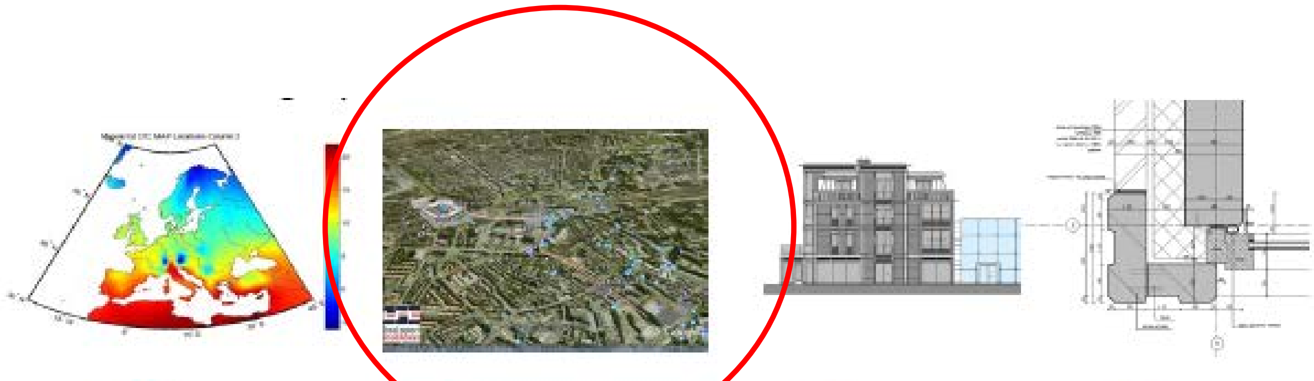


Scale level [m] Building Physics
Indoor climate performance & design

Scale levels, from left to right: EU; Urban area; Building; Material;



Urban Scale



Scale levels, from left to right: EU; Urban area; Building; Material;

- **[km] Urban Physics**

Methodology

- Create random generated urban area using Matlab/Comsol
- Add turbulence model & boundary values
- Physics controlled meshing & solve
- Check solution regarding wall functions (improve mesh if necessary)
- Check final solution

Methodology

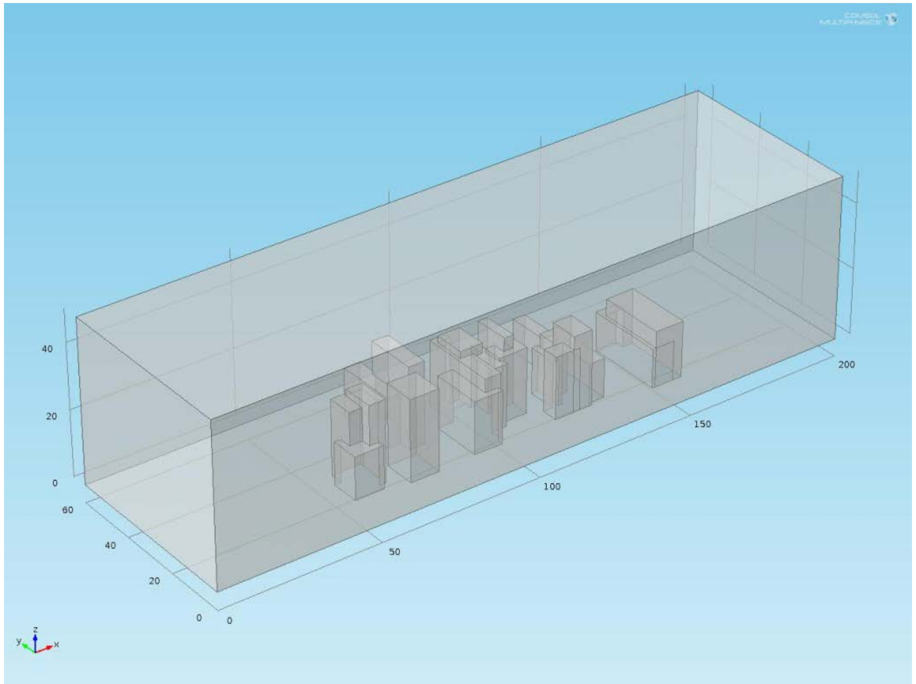
- **Create random generated urban area using Matlab/Comsol**
- Add turbulence model & boundary values
- Physics controlled meshing & solve
- Check solution regarding wall functions (improve mesh if necessary)
- Check final solution

Technische Universiteit Eindhoven University of Technology

```
import com.comsol.model.*
import com.comsol.model.util.*
model = ModelUtil.create('Model');
model.modelPath('D:\COMSOL42\mfiles');
model.name('MultiBuildingGeo1.mph');
model.modelNode.create('mod1');

model.geom.create('geom1', 3);
model.geom('geom1').feature.create('wp1', 'WorkPlane');
model.geom('geom1').feature('wp1').geom.feature.create('r1', 'Rectangle');
model.geom('geom1').feature.create('ext1', 'Extrude');
model.geom('geom1').feature('wp1').geom.feature('r1').set('size', {'2000' '600'});
model.geom('geom1').feature('ext1').setIndex('distance', '100', 0);
model.geom('geom1').feature('ext1').selection('input').set({'wp1.r1'});

nB=80;
xp=100+400*rand(nB,1);
yp=500+500*rand(nB,1);
sx=2+8*rand(nB,1);
sy=5+20*rand(nB,1);
hz=10+20*rand(nB,1);
```



Methodology

- Create random generated urban area using Matlab/Comsol
- **Add turbulence model & boundary values**
- Physics controlled meshing & solve
- Check solution regarding wall functions (improve mesh if necessary)
- Check final solution

Technische Universiteit **Eindhoven** University of Technology

Spalart-Allmaras

Turbulent Flow, Spalart-Allmaras

Selection

Geometric entity level	Domain
Selection	Domain 1

Equations

$$\rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot\left[-p\mathbf{I} + (\mu+\mu_T)\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) - \frac{2}{3}(\mu+\mu_T)(\nabla\cdot\mathbf{u})\mathbf{I}\right] + \mathbf{F}$$

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

$$(\mathbf{u}\cdot\nabla)\nu^t = C_{b1}S^t\nu^t - C_{w1}f_w\left(\frac{\nu^t}{l_w}\right)^2 + \frac{1}{\sigma_{\tilde{\nu}}}\nabla\cdot((\nu+\nu^t)\nabla\nu^t) + \frac{C_{b2}}{\sigma_{\tilde{\nu}}}\nabla\nu^t\cdot\nabla\nu^t, \quad \nu^t = \text{nutilde}$$

$$\nabla G\cdot\nabla G + \sigma_w G(\nabla\cdot\nabla G) = (1+2\sigma_w)G^4, \quad l_w = \frac{1}{G} - \frac{l_{ref}}{2}$$

$$\mu_T = \rho\nu^t f_{v1}, \quad C_{w1} = \frac{C_{b1}}{\kappa_v} + \frac{1+C_{b2}}{\sigma_{\tilde{\nu}}}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}}, \quad \chi = \frac{\nu^t}{\nu}$$

$$f_w = g\left(\frac{1+C_{w3}^6}{g^6 + C_{w3}^6}\right)^{1/6}, \quad g = r + C_{w2}(r^6 - r), \quad r = \frac{\nu^t}{S^t\kappa_v^2 l_w^2}$$

$$S^t = S + \frac{\nu^t}{\kappa_v^2 l_w^2}f_{v2}, \quad S = \sqrt{2\mathbf{\Omega}:\mathbf{\Omega}}, \quad \mathbf{\Omega} = \frac{1}{2}(\nabla\mathbf{u} - (\nabla\mathbf{u})^T)$$

Table 1 Wind profile at inlet

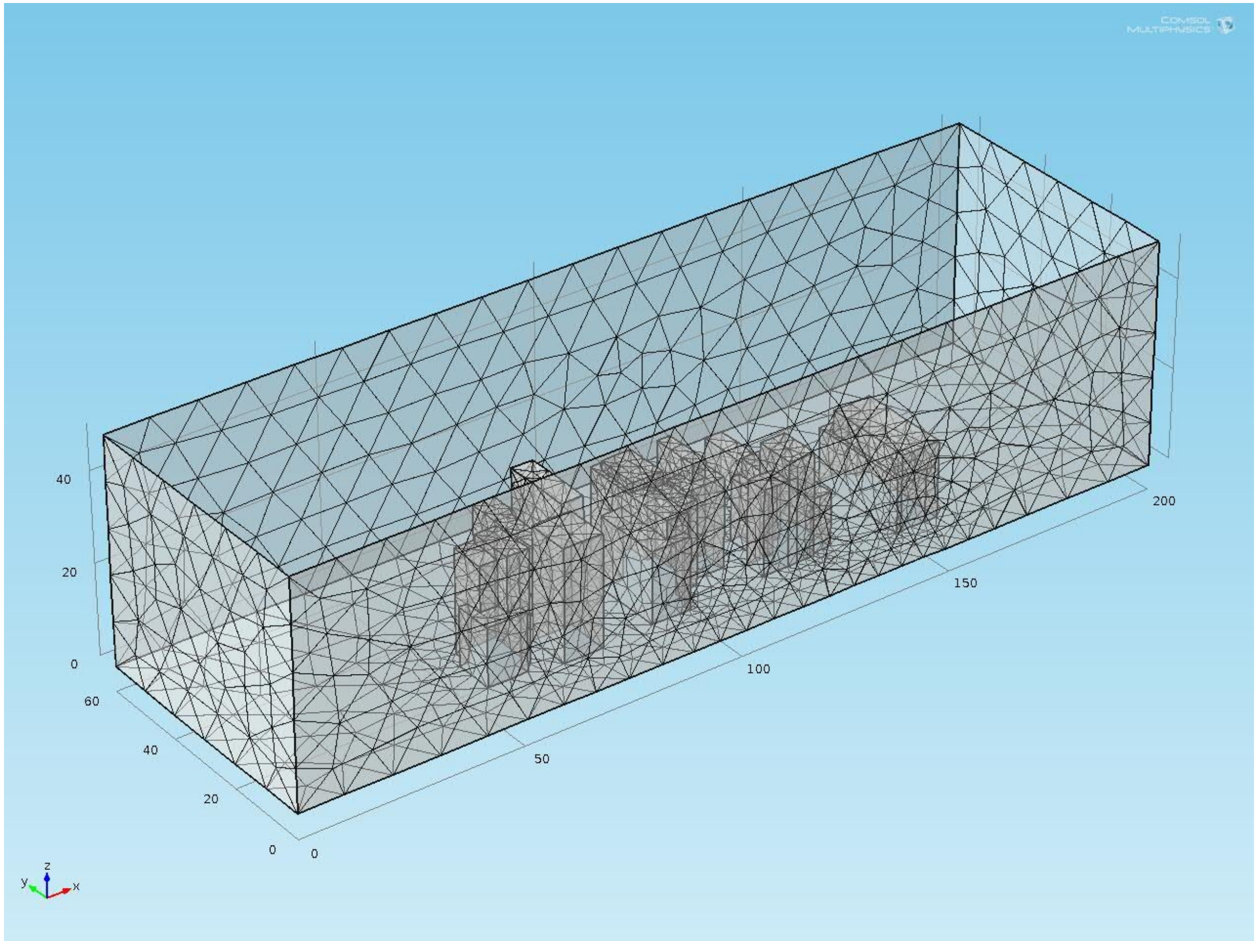
Description	Value
Normal inflow velocity	(0.912/0.41)*log((z + 0.1)/0.1)

Methodology

- Create random generated urban area using Matlab/Comsol
- Add turbulence model & boundary values
- **Physics controlled meshing & solve**
- Check solution regarding wall functions (improve mesh if necessary)
- Check final solution

Technische Universiteit Eindhoven University of Technology

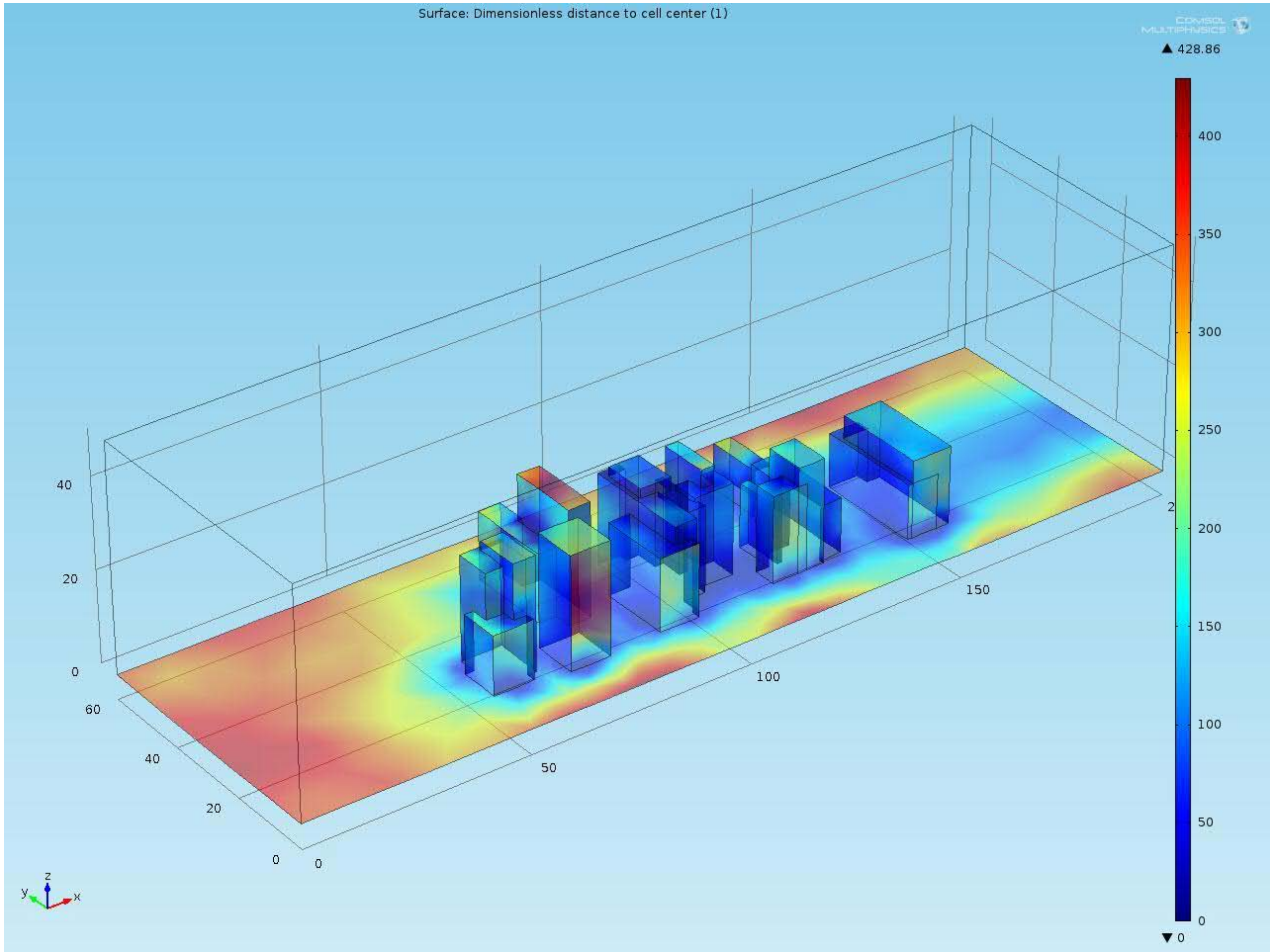
Spalart-Allmaras



Methodology

- Create random generated urban area using Matlab/Comsol
- Add turbulence model & boundary values
- Physics controlled meshing & solve
- **Check solution regarding wall functions (improve mesh if necessary)**
- Check final solution

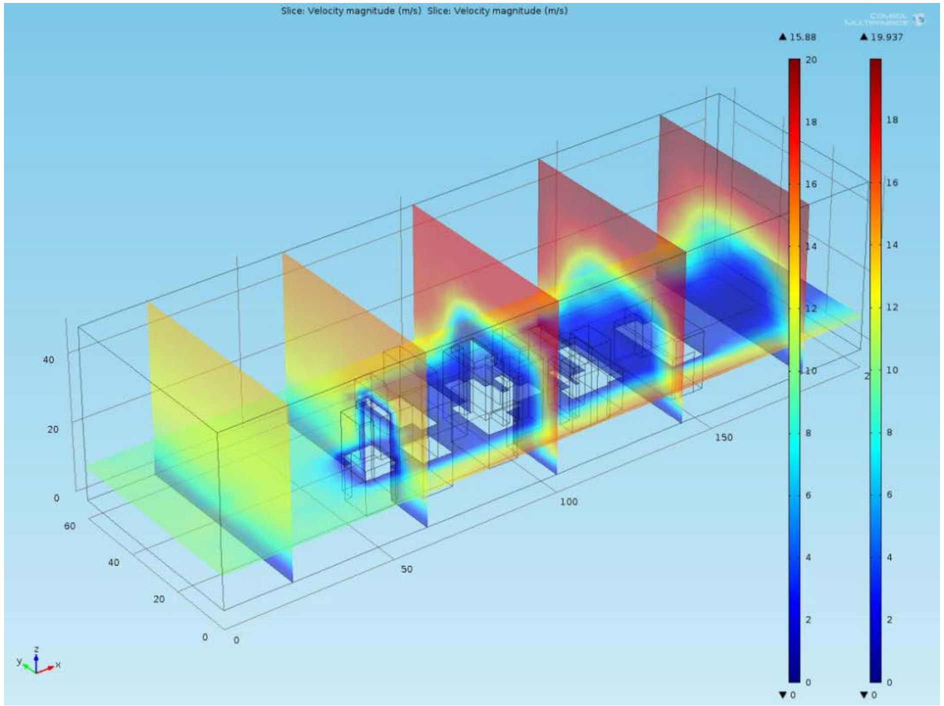
Spalart-Allmaras

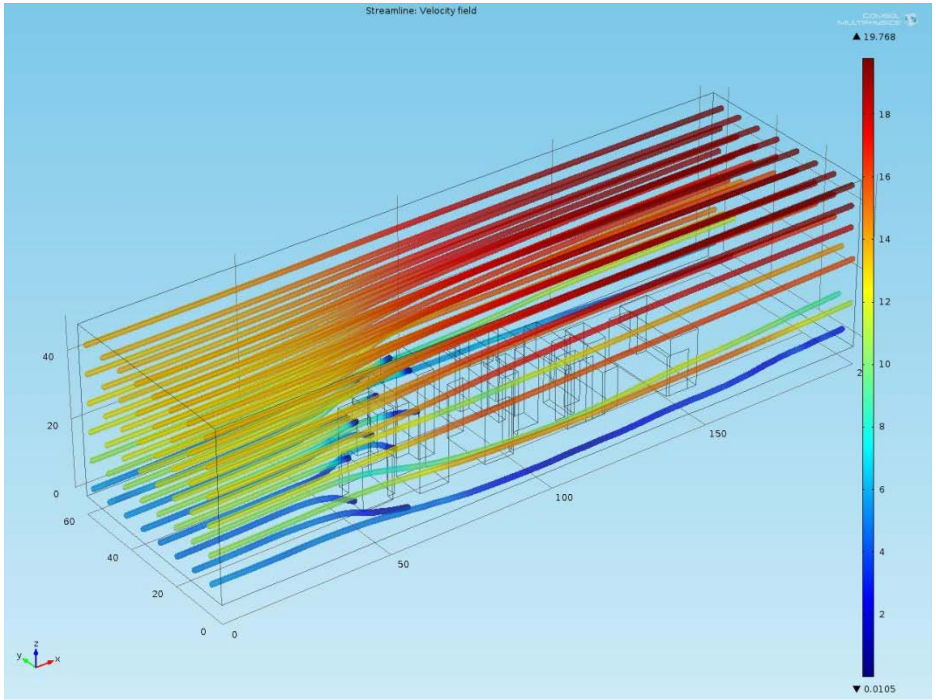


Methodology

- Create random generated urban area using Matlab/Comsol
- Add turbulence model & boundary values
- Physics controlled meshing & solve
- Check solution regarding wall functions (improve mesh if necessary)
- **Check final solution**

Spalart-Allmaras





K-eps Turbulence model

Turbulent Flow, k-ε

Selection

Geometric entity level	Domain
Selection	Domain 1

Equations

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

$$\nabla\cdot\left[-p\mathbf{I} + (\mu+\mu_T)\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) - \frac{2}{3}(\mu+\mu_T)(\nabla\cdot\mathbf{u})\mathbf{I} - \frac{2}{3}\rho k\mathbf{I}\right] + \mathbf{F}$$

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

$$\rho(\mathbf{u}\cdot\nabla)k = \nabla\cdot\left[\left(\mu + \frac{\mu_T}{\sigma_k}\right)\nabla k\right] + P_k - \rho\epsilon$$

$$\rho(\mathbf{u}\cdot\nabla)\epsilon = \nabla\cdot\left[\left(\mu + \frac{\mu_T}{\sigma_\epsilon}\right)\nabla\epsilon\right] + C_{\epsilon 1}\frac{\epsilon}{k}P_k - C_{\epsilon 2}\rho\frac{\epsilon^2}{k}, \quad \epsilon = \mathrm{ep}$$

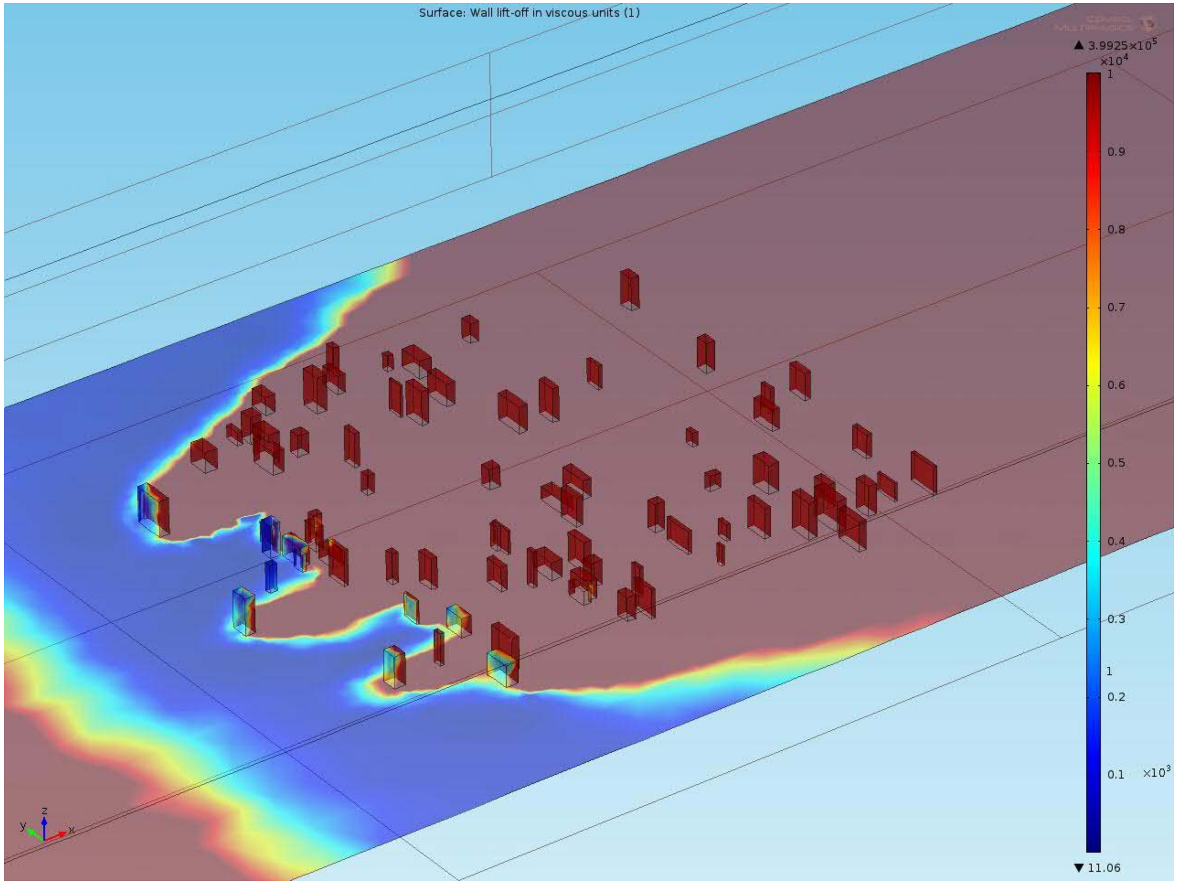
$$\mu_T = \rho C_\mu \frac{k^2}{\epsilon}$$

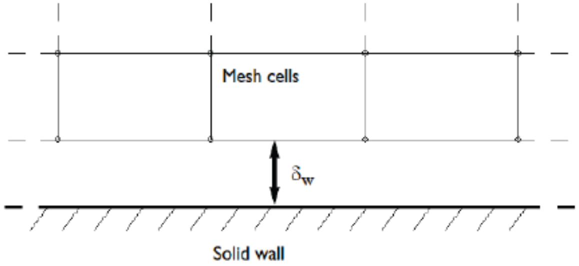
$$P_k = \mu_T\left[\nabla\mathbf{u} : \left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) - \frac{2}{3}(\nabla\cdot\mathbf{u})^2\right] - \frac{2}{3}\rho k\nabla\cdot\mathbf{u}$$

Table 1 Wind profile at inlet

Description	Value
Normal inflow velocity	(0.912/0.41)*log((z + 0.1)/0.1)

K-eps Turbulence model

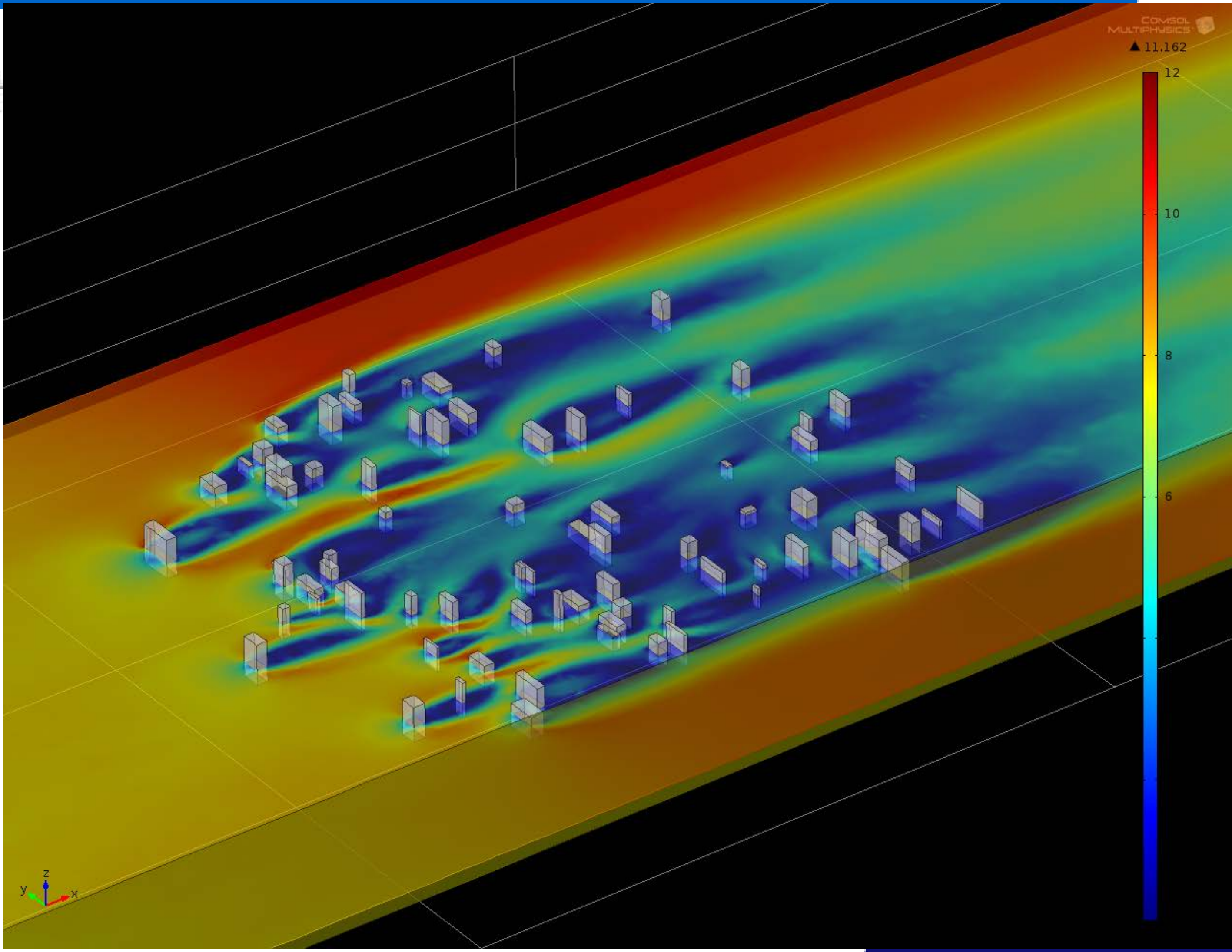




The distance δ_w is automatically computed so that $\delta_w^+ = \rho u_\tau \delta_w/\mu$, where $u_\tau = C_\mu^{1/4}\sqrt{k}$ is the friction velocity, becomes 11.06.

Scale level [km] Urban physics
Urban climate performance

Scale levels, from left to right: EU; Urban area; Building; Material;



Conclusions

- **Scales with successful simulation (so far):**
  - **Spalart-Allmaras: ~0.1km x 0.1km x 0.05km**
  - **K- eps: ~0.6km x 2km x 0.1km**

- **Future research:**
  - **Limits ?**