Comparing strategies for pedestrian wind comfort and safety around high-rise buildings

Jacob Wahlgren

KTH

2021-06-06

Bridgewater Place, UK

- Strong winds caused by high-rise buildings
- Lethal accident by overturned truck in 2011

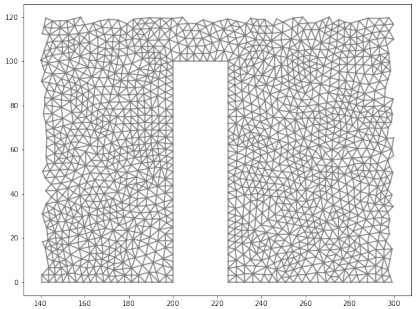
Important factors

- Height
- Shape
- Angle to the wind
- Surrounding buildings
- Trees and shrubbery
- Fences and podiums

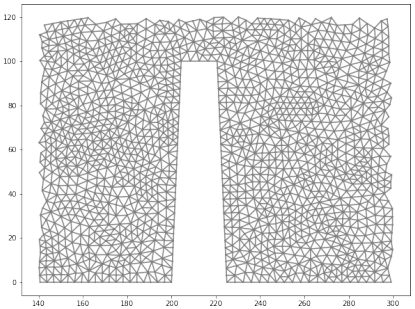
Research question

What are the most effective designs for high-rise buildings to ensure pedestrian wind comfort and safety?

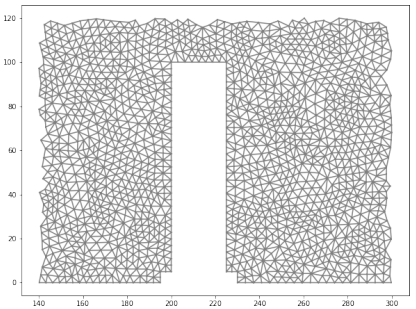
Scenarios



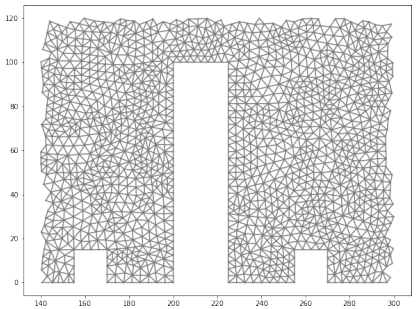
(a) standard



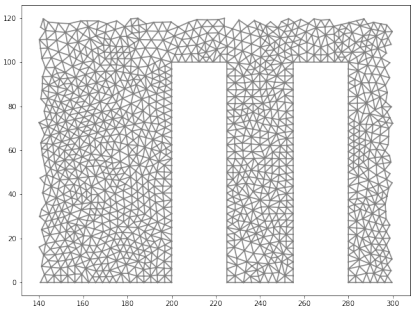
(b) tapered



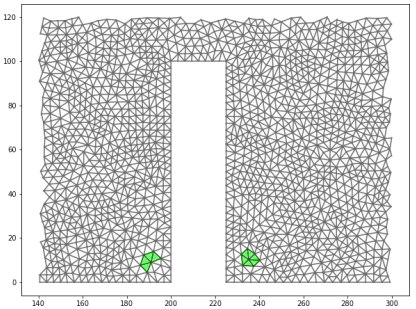
(c) podium



(d) neighbors



(e) twins



(f) trees

Measurements

- Wind speed at 1.5 m height
- Drag force on buildings
- Time period of 120 s

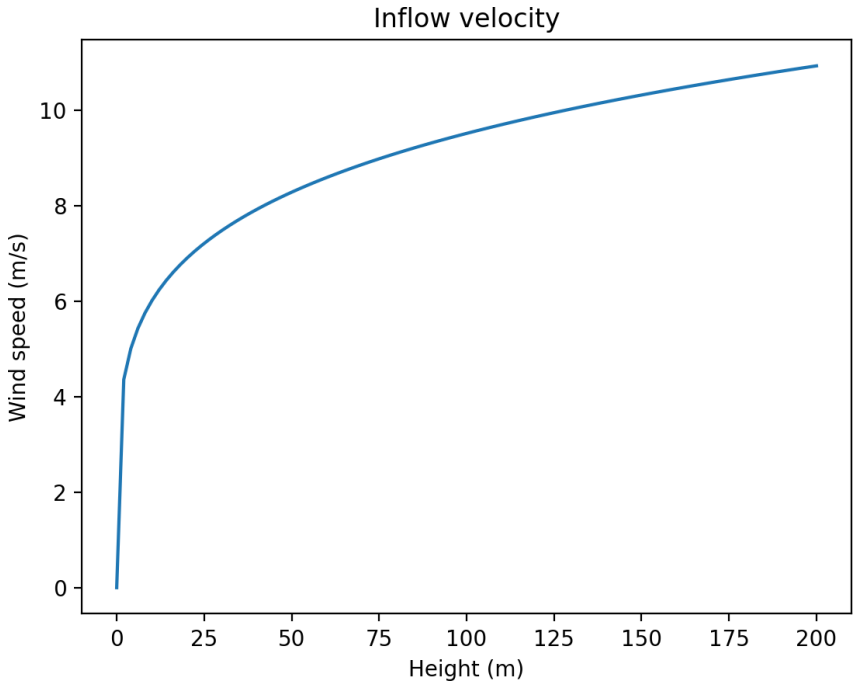
CFD method

- Based on DFS from "Towards a parameter free-method for high Reynolds number turbulent flow simulation..."
- Building free-slip boundaries using skin friction penalty term
- Porous media using Darcy drag term from Brinkman model

Boundary conditions

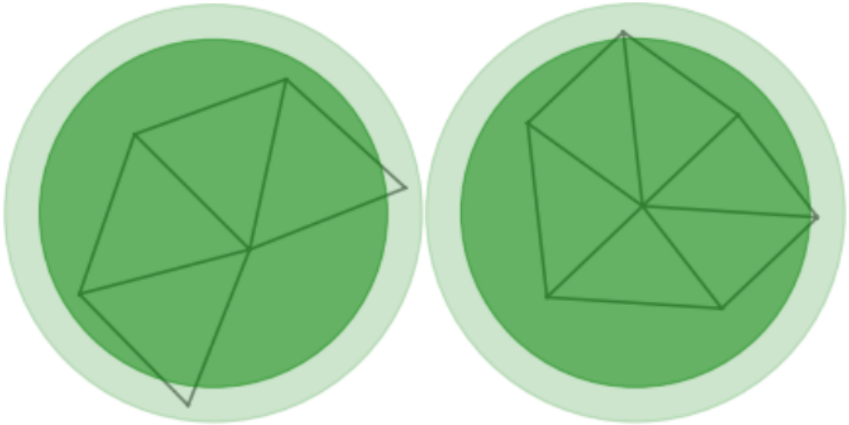
Boundary	Condition	Formula
Ground	No slip	$u = 0$
Sky	Free slip	$(u, n) = 0$
Building walls	Free slip	$(u, n) = 0$
Inflow	Velocity profile	$u = 6 \cdot (\frac{y}{10})^{0.2}$
Outflow	Zero pressure	$p = 0$

Inflow profile



- Power law in height common model
- Moderate wind (6 m/s at 10 m height)

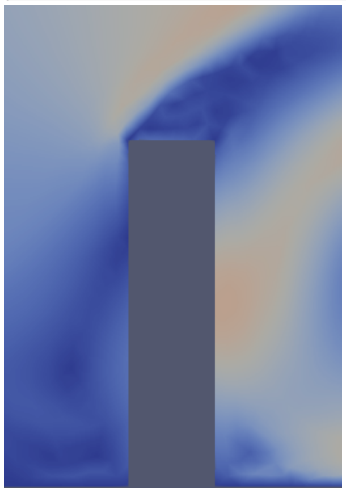
Porous media domain



```
class PorousDomain(SubDomain):
    def inside(self, x, on_boundary):
        for zx, zy, zr in porous_zones:
            zr += 1
            if between(np.sqrt((x[0]-zx)**2 + (x[1]-zy)**2), (0, zr)):
                return True
        return False

porous_domain = PorousDomain()

domains = MeshFunction("size_t", mesh, mesh.topology().dim())
domains.set_all(0)
porous_domain.mark(domains, 1)

dx_sub = Measure("dx")(subdomain_data=domains)
```

Porous media equation

```
Fu = inner((u - u0)/dt + grad(um)*um1, v)*dx - p1*div(v)*dx + nu*inner(gra
    + d1*inner((u - u0)/dt + grad(um)*um1 + grad(p1), grad(v)*um1)*dx + d2
    + alpha*(inner(dot(um,normal), dot(v,normal)))*ds(5) + beta*(inner(dot
    + nu*Kinv_diag*(inner(um[0],v[0]) + inner(um[1],v[1]))*dx_sub(1)
```

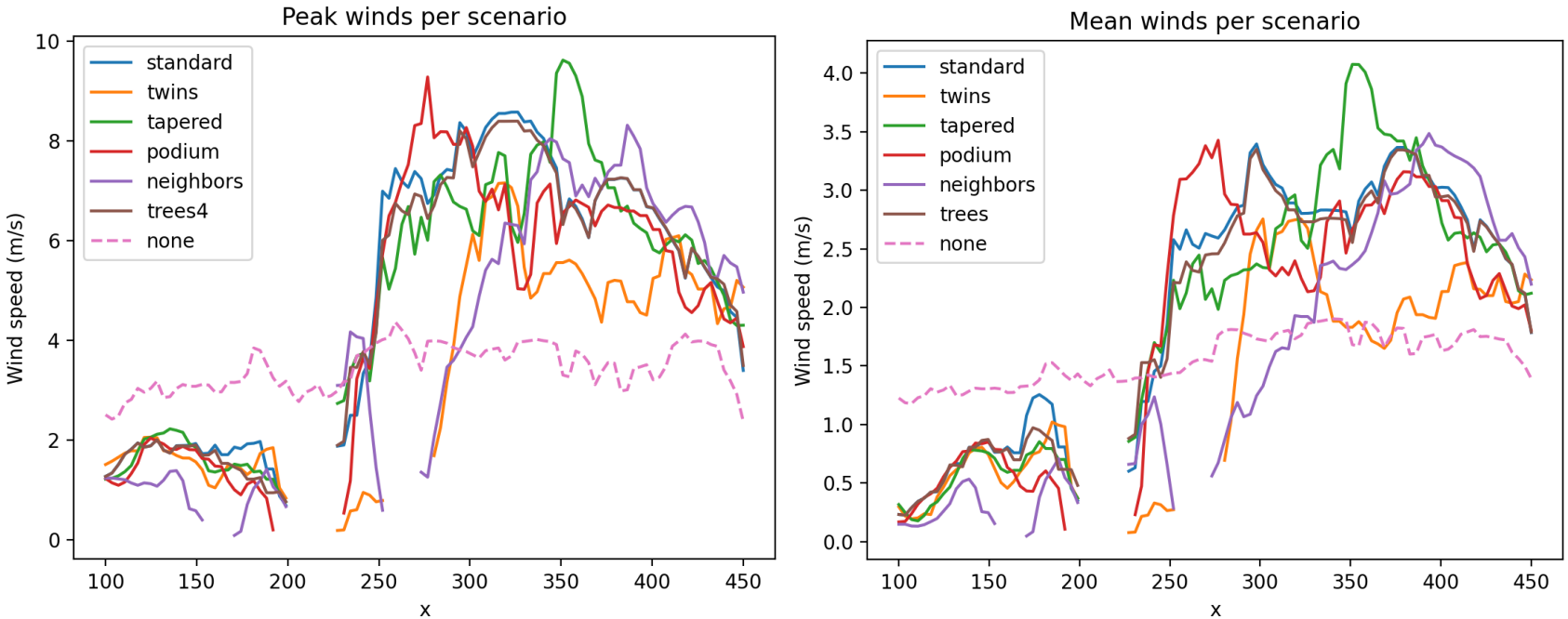


$$K = 10^{-4}$$

Flow pattern

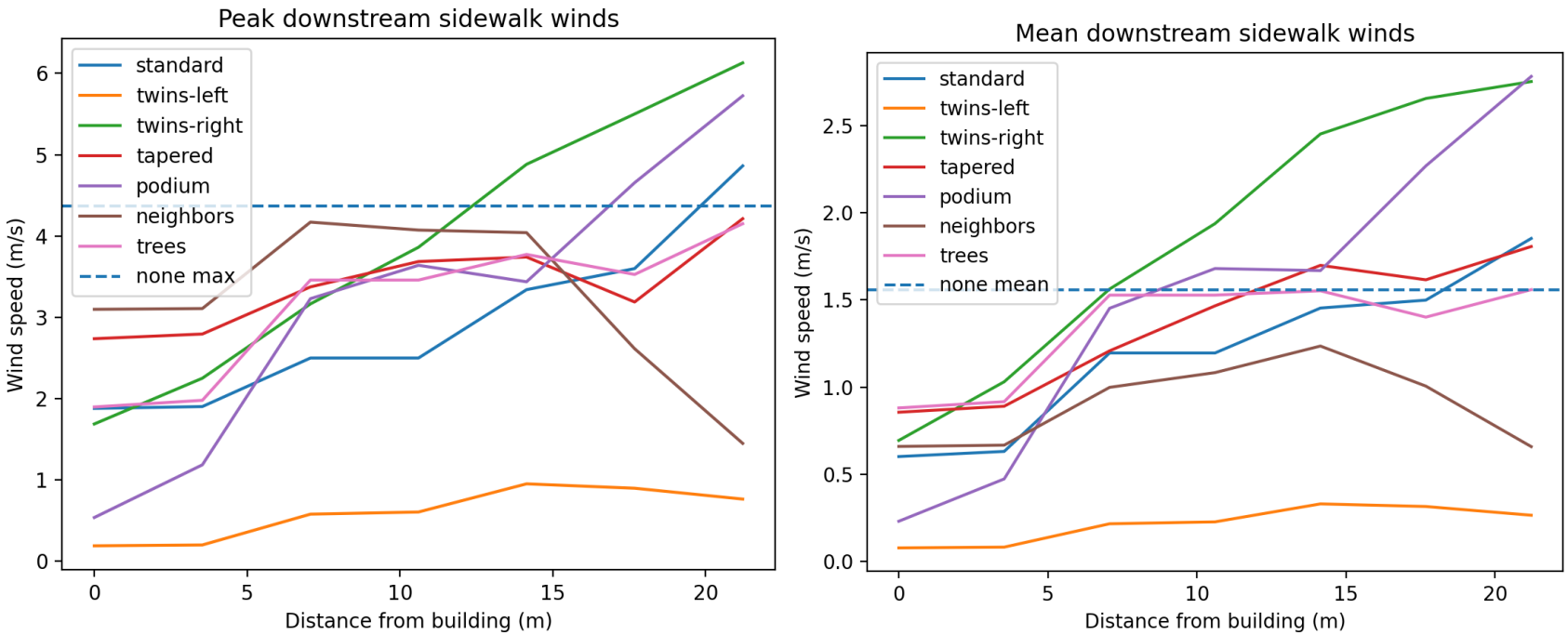
See separate animations.

Pedestrian wind speed overview



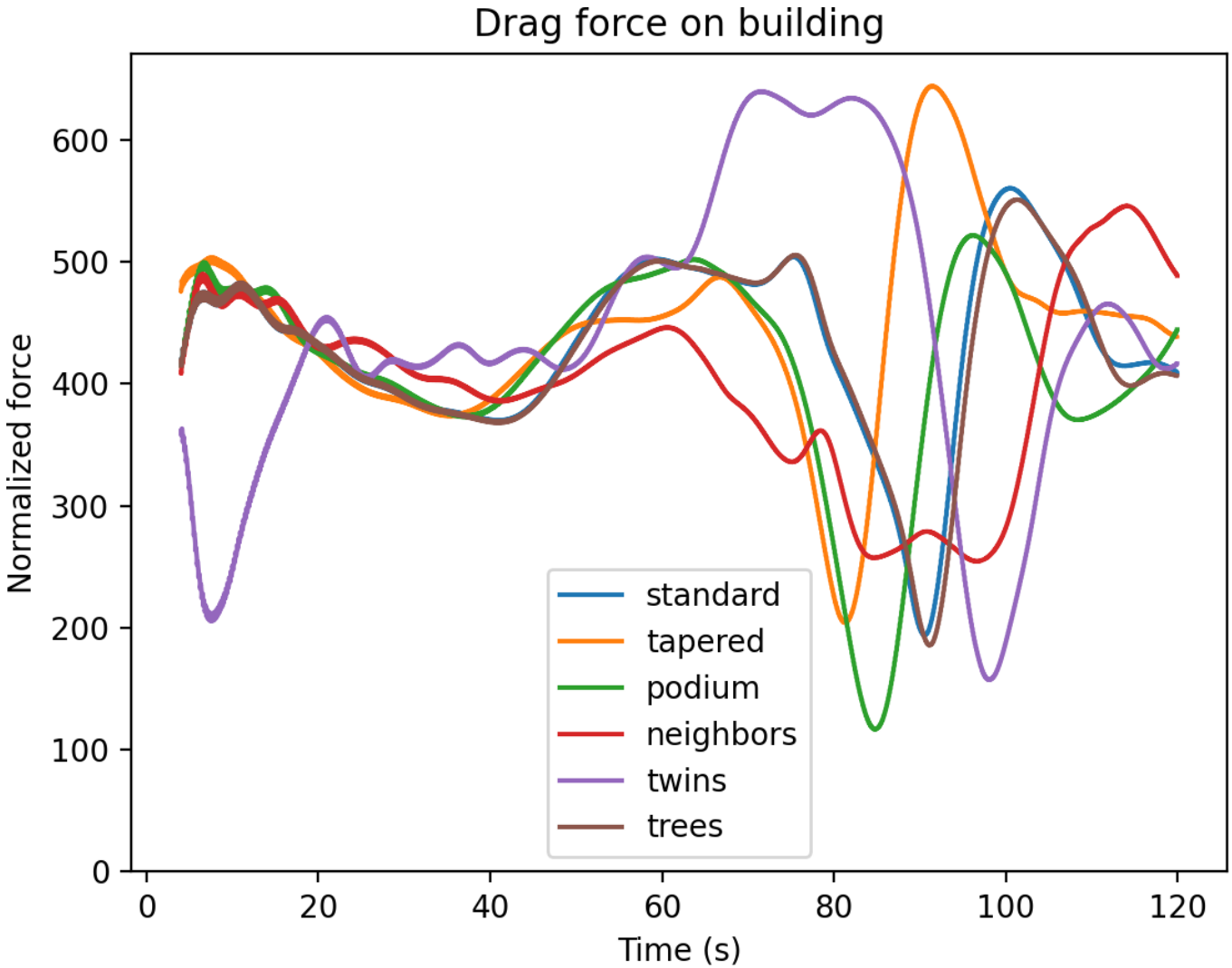
- Decrease upstream, increase downstream
- Neighbors stands out

Pedestrian wind speed on downstream sidewalk



- Left twin lowest, right twin highest
- Podium is low close to building

Drag force


Drag force on building
Normalized force
Time (s)
0
100
200
300
400
500
600
0
20
40
60
80
100
120
standard
tapered
podium
neighbors
twins
trees

Analysis

- Similar results for standard, tapered and trees
- Twins had low winds between, but high downstream
- Neighbors had lower drag and overall wind speeds
- Podium had lower wind speeds within 5 m
- Correlation between wind speed and drag

Limitations of 2D model

- 3D turbulence effects
- Street canyon effect for twins and neighbors

Time period issues

- Observe oscillation to ensure stable state
- Divergence after 120 s
- Tentative: 200+ s possible by making timestep 7 times shorter (over 24 h simulation time)

Conclusion

- Best strategy based on results was neighboring low-rise buildings, or several high-rise buildings together
- Lead to reduced wind speed and drag
- Refined experiments required to confirm

Future work

- Increased mesh resolution and shorter timestep
- Adaptive mesh refinement
- Use free-slip condition for ground boundary
- 3D model
- Use wind comfort model
- Wind tunnel experiments

Thank you!

Questions?