Damage detection of concrete slab bridges

Jeremy

December 12, 2019

Dutch Main Road Network

Stucture type	Number	Deck Area (m2)
Concrete bridge	3'131	3'319'002
Steel bridge (fixed)	88	301'997
Movable bridge	43	347'876
Tunnel	14	475'228
Aqueduct	7	86'491
Total	3'283	4'530'593

Existing SHM

- State of the art do not determine extent of damage
- If they do:
  - assumptions are made
  - or in a controlled experiment
- Only determine whether damage present or not
  - global health monitoring
- Detection only necessary to then take further action
  - on-site examination

Ultimate Goal

- Damage detection of concrete slab bridges
- Major problem:
  - data of damaged state not available
    - can't go around damaging bridges
- Then how to prototype damage detection systems without data?

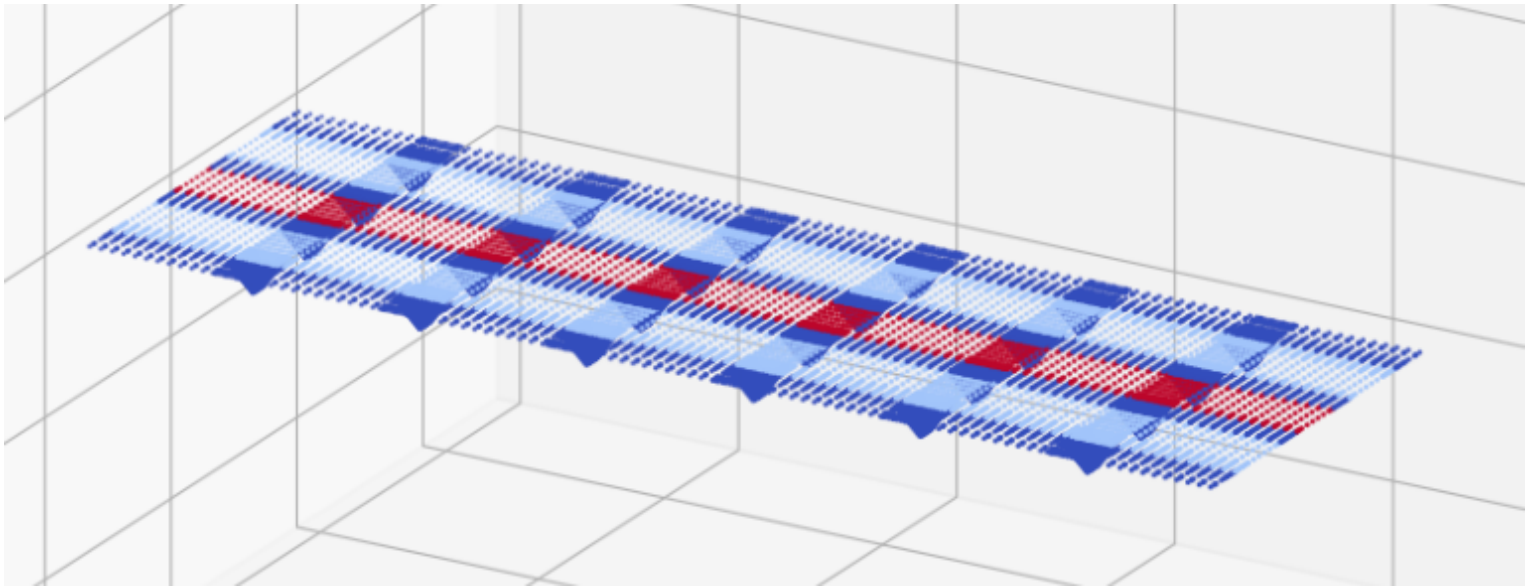
Ultimate Goal

- Damage detection of concrete slab bridges
- Based on simulated data
- Validated against real data
- System can be applied to many concrete bridges
  - model is parametric
  - not just for one bridge

Bridge Model: Parametric

```
data Bridge {
  length          :: Float,
  width           :: Float,
  piers           :: [Pier],
  deck_materials  :: [Material],
  lanes           :: [Lane],
}
```

Generated Model: Bridge 705



Generated Model

- `Bridge` model is parametric
  - material propertes, dimensions
- can generate 2D beam or 3D shell model

Mesh: Deck Nodes

- base mesh = 50 * 20
- from piers = 12 * 68
- from loads = 0 * 0
- from materials = 0 * 24
- from pier refinement = 66 * 0
- total = 128 * 112

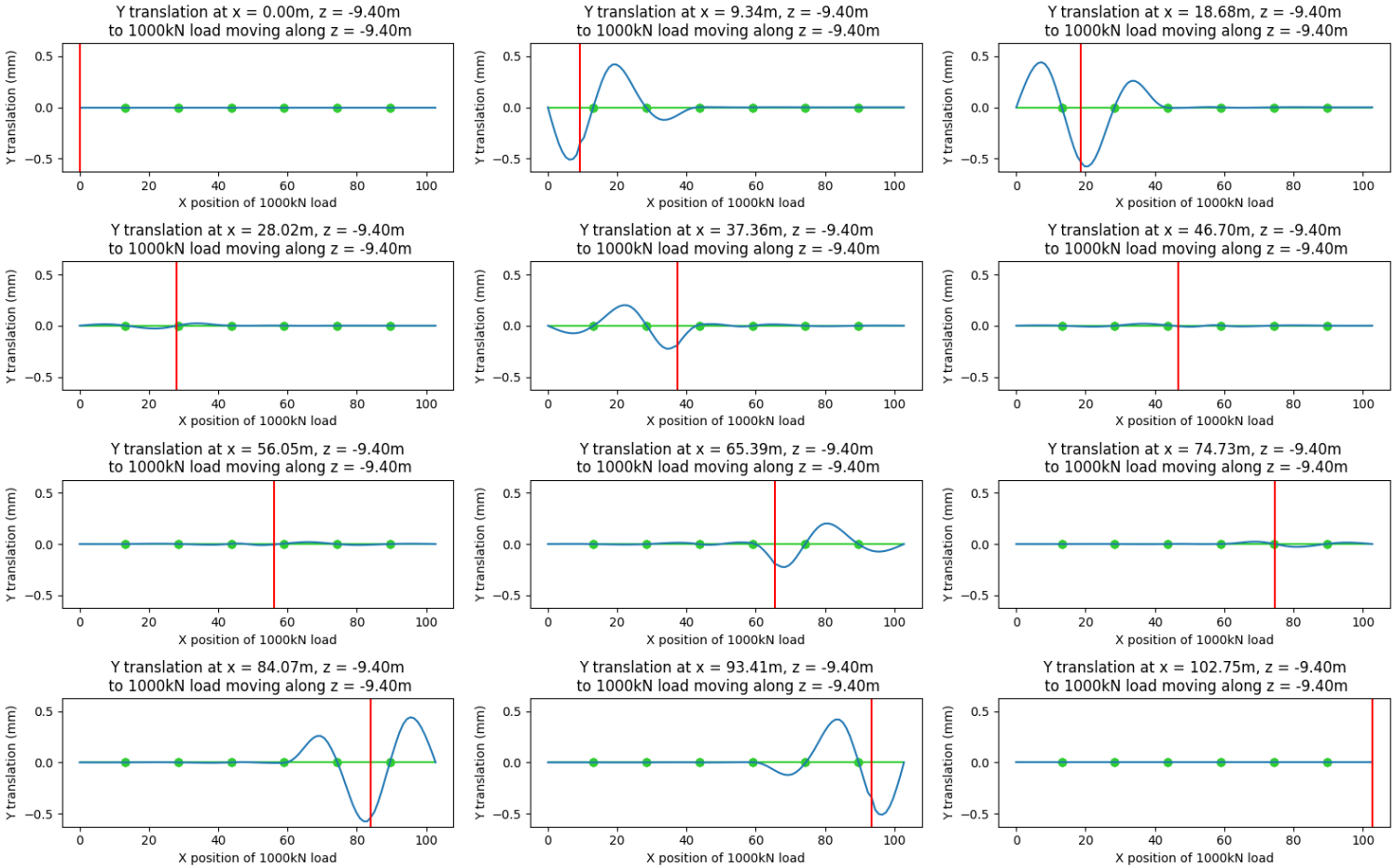
Mesh: Pier Nodes

- base mesh = 17 * 17
- from deck = 0.0 * 7.0 (mean)
- total = 17 * 24

Animation

./animation.mp4

Unit Load Simulations



Responses via Superposition: Traffic

- 2 lanes, 4 wheel tracks
- vehicle leaving on bottom lane
- lighter vehicle leaving top lane

```
traffic_at_time_0 = [
    1, 2, 0,
    1, 2, 0,
    0, 3, 3,
    0, 3, 3]
```

Responses via Superposition: Traffic

- `t` rows, one per unit time
- `n * 4` columns, n per wheel track
- currently using `n=100` (Bridge 705 = 102.75m)
- TODO: `n = 102.75 / wheel print length`

```
traffic = [
    [1, 2, 0, 1, 2, 0, 0, 3, 3, 0, 3, 3],
    [2, 0, 0, 2, 0, 0, 0, 0, 3, 0, 0, 3],
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
```

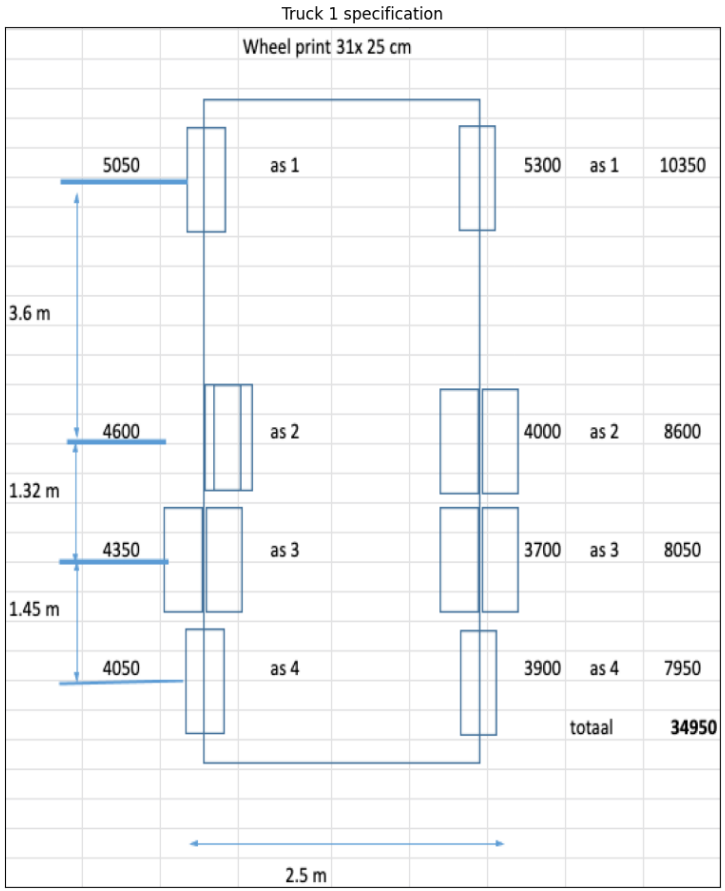
Responses via Superposition: Unit Load Simulations

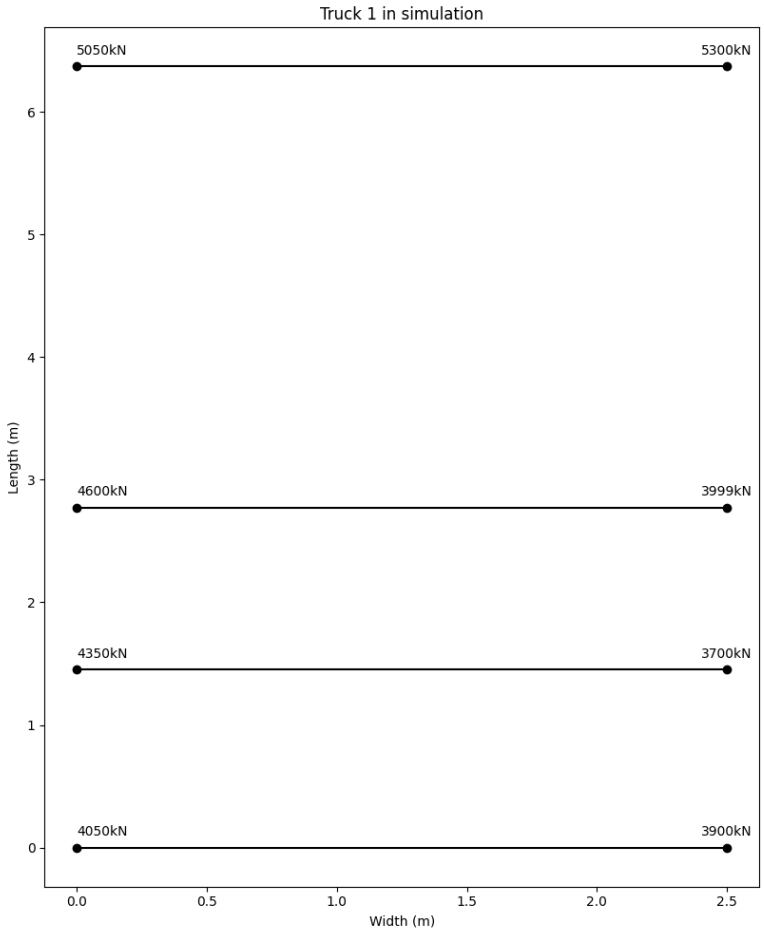
- `n * 4` rows, n per wheel track
- `p` columns, points you want a response at
- Each cell is
  - the response at point p_i
  - from unit load simulation `j`

```
points = [
    [1, 0, 0, 2]
    [1, 0, 0, 3]
    [2, 1, 0, 3]
    [2, 1, 0, 2]
    [1, 2, 1, 2]
    [1, 2, 1, 1]
    [0, 1, 0, 1]
    ...
```

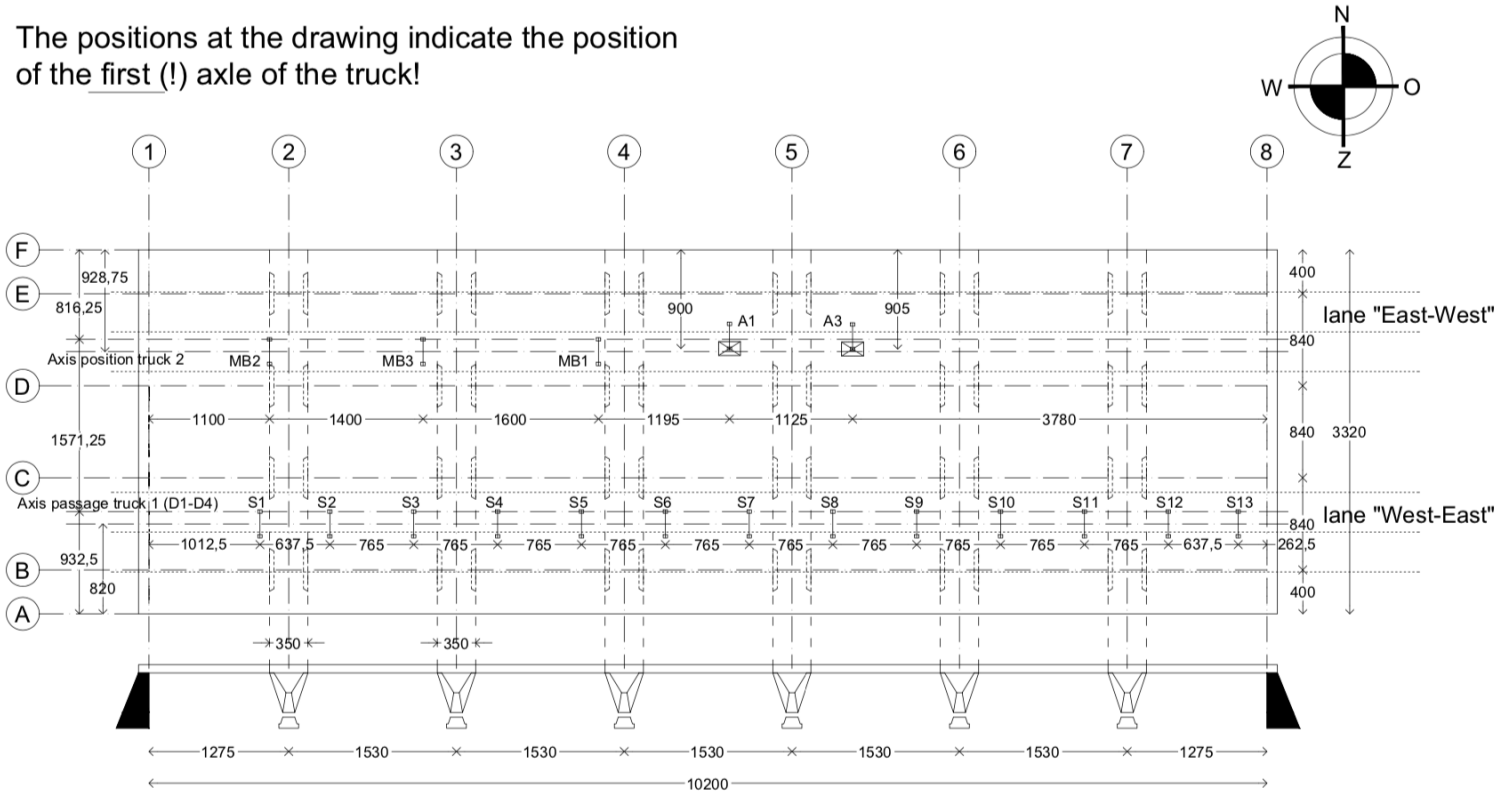
Validation: Truck 1 in experimental campaign

fix plot

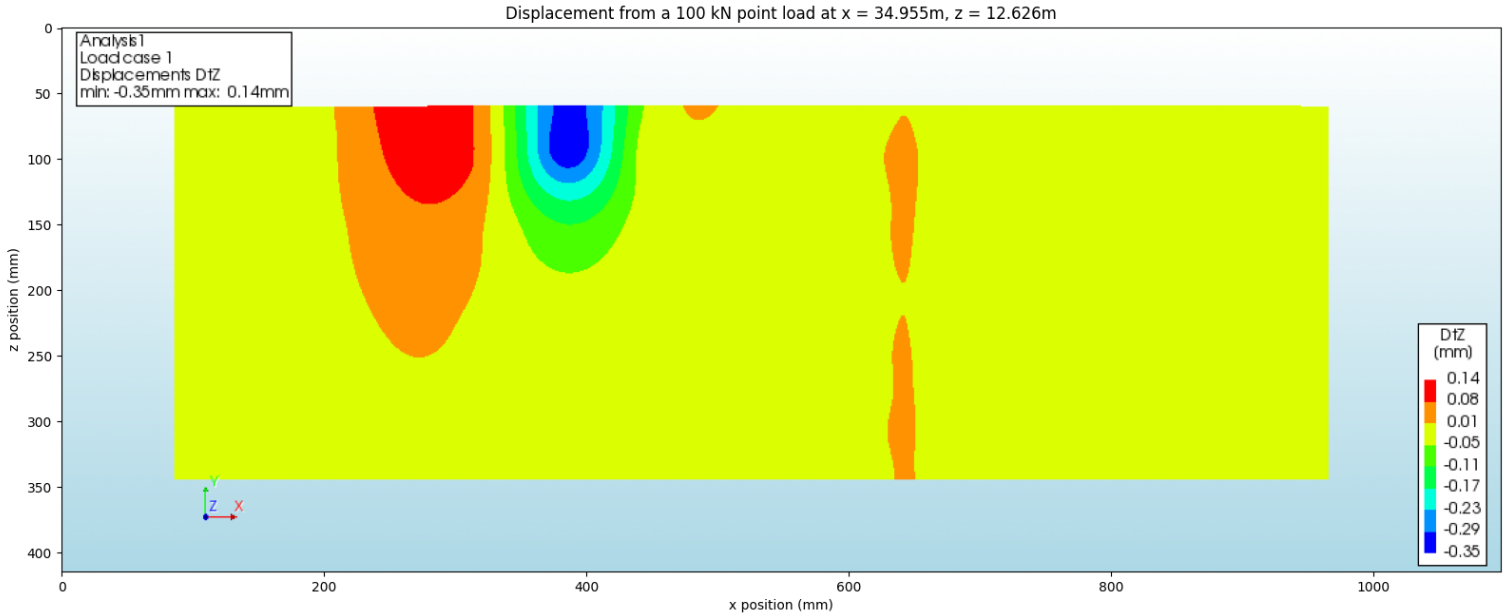




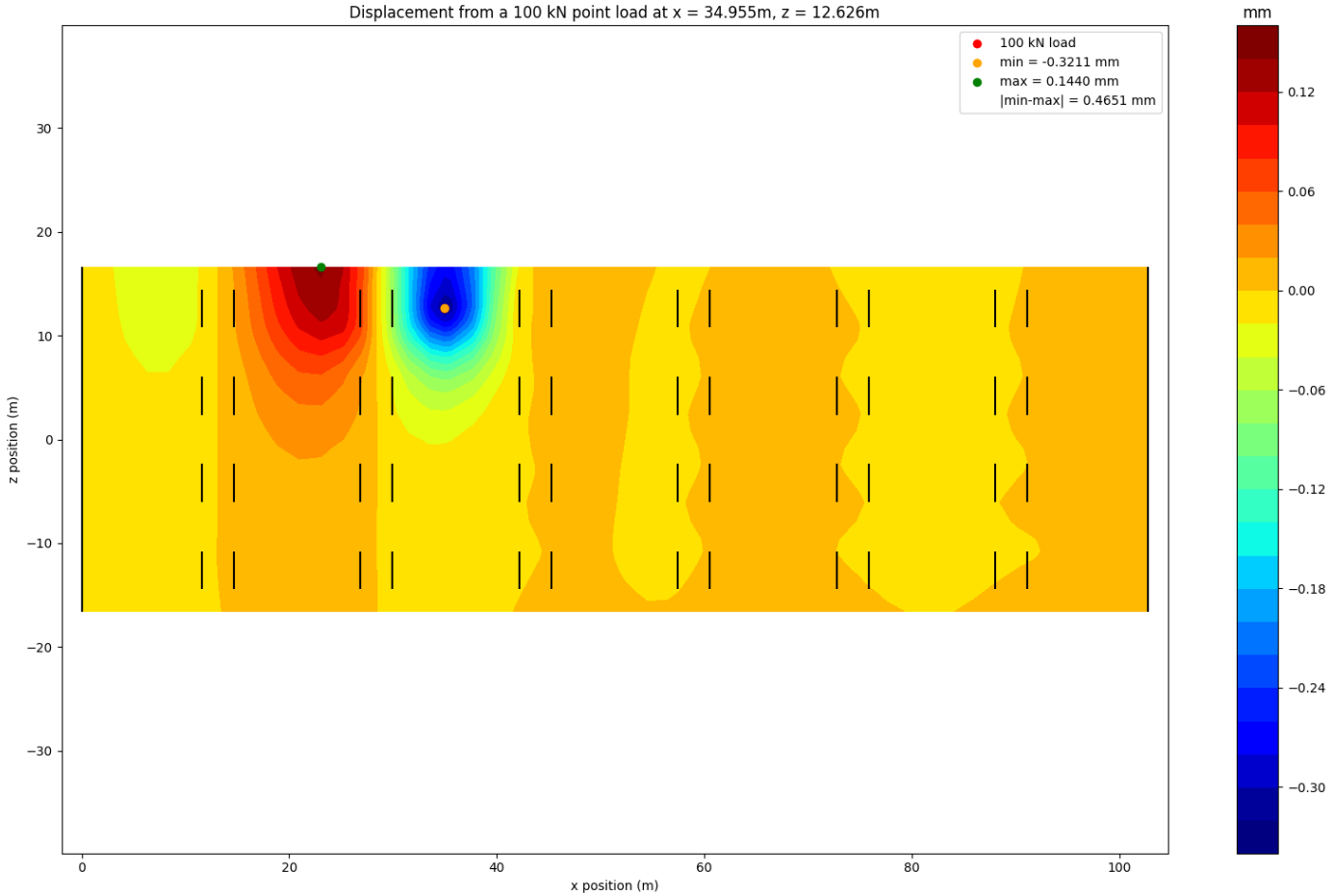
Validation: Truck positions in Experimental Campaign



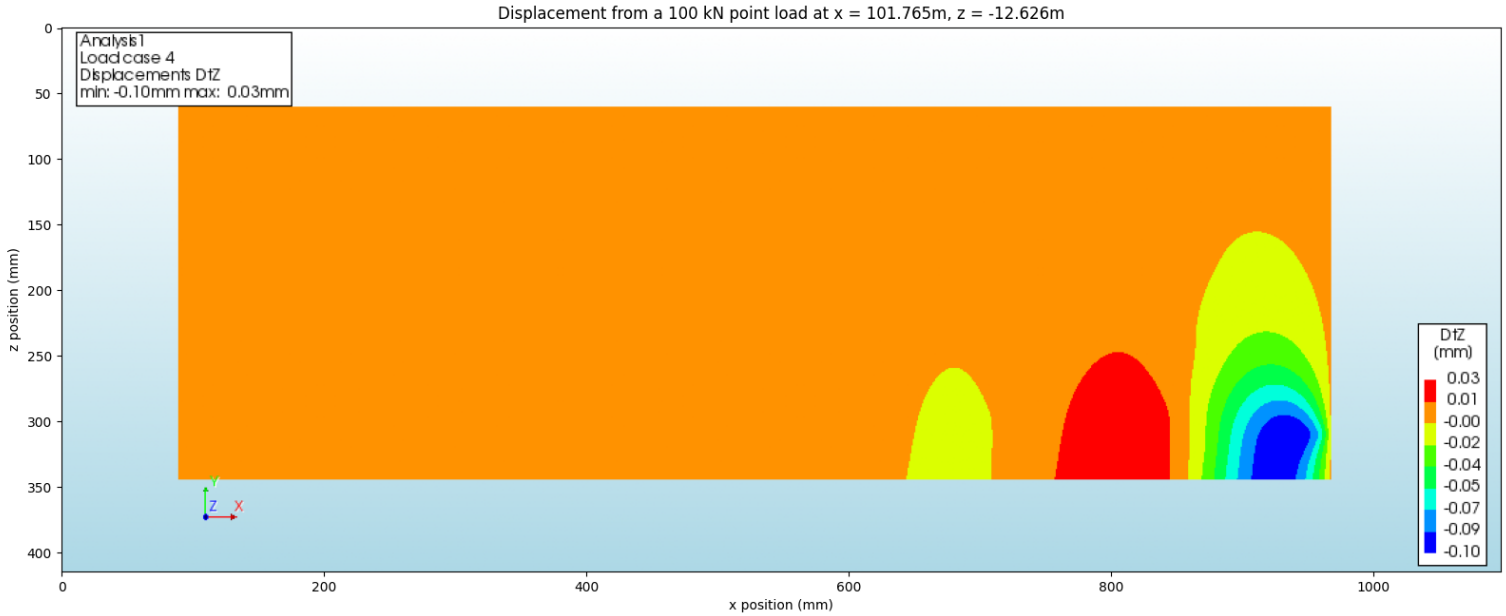
Validation: Displacement Diana



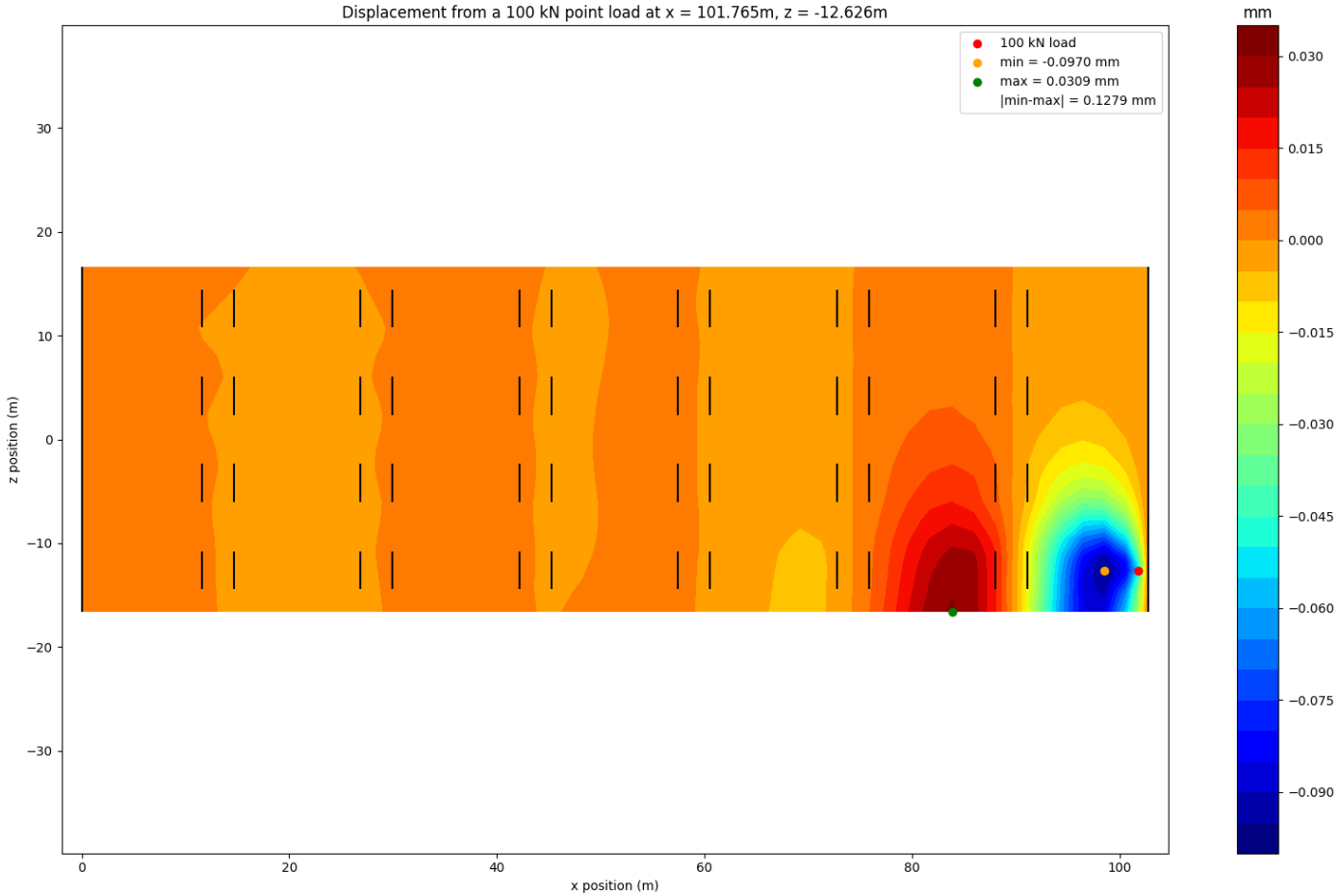
Validation: Displacement OpenSees



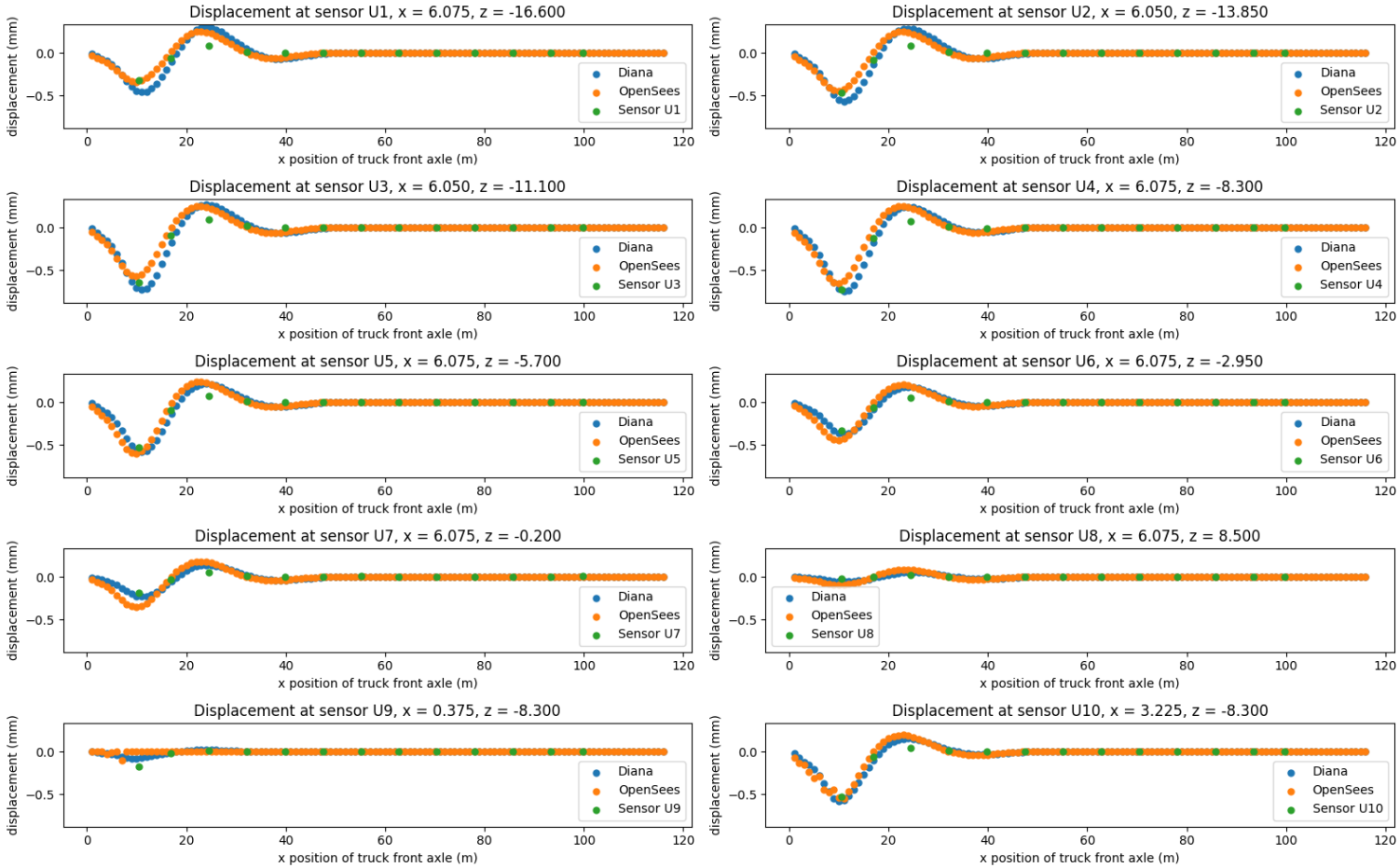
Validation: Displacement Diana



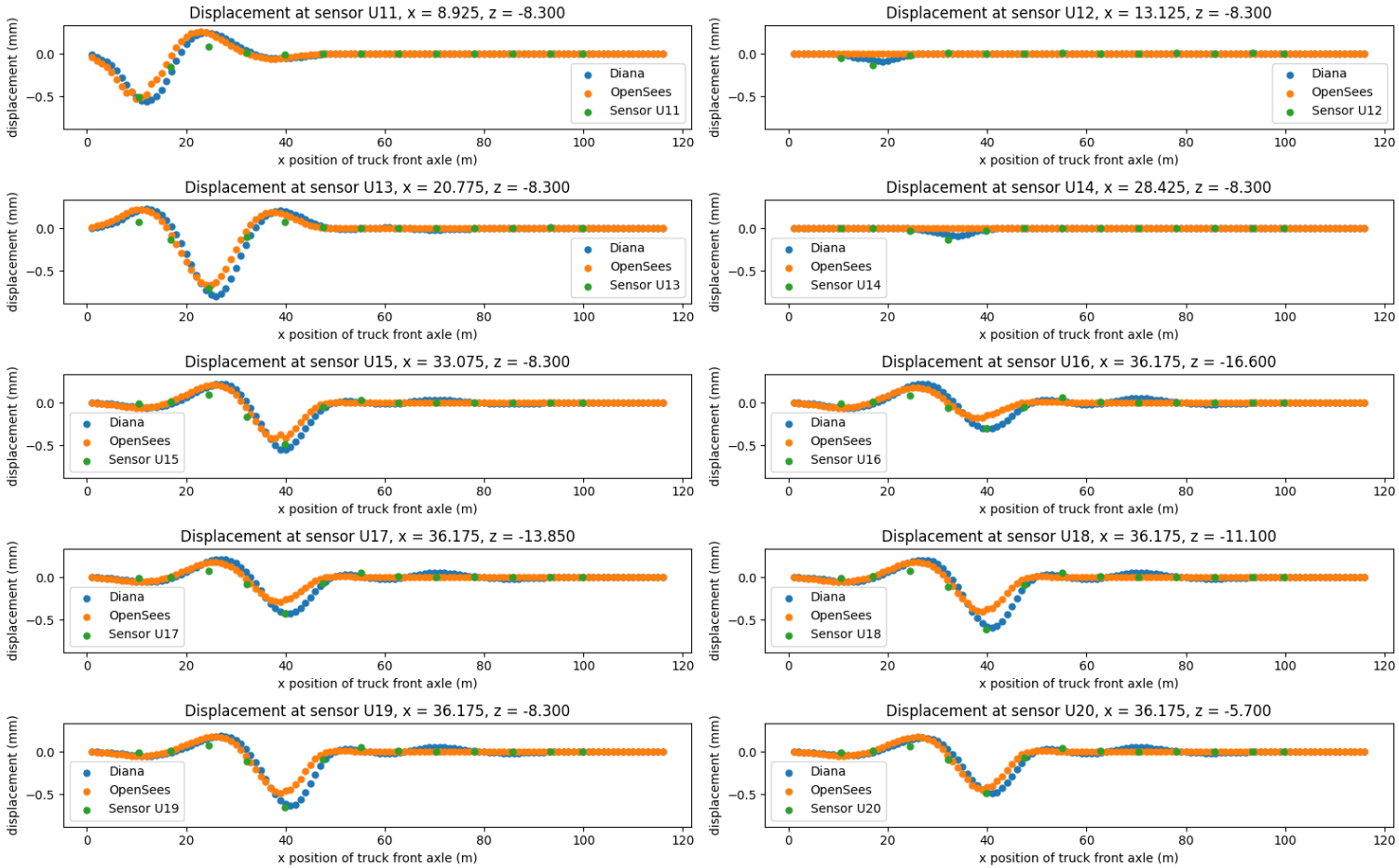
Validation: Displacement OpenSees



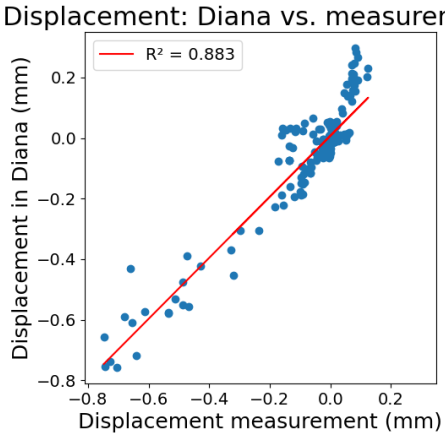
Validation: Displacement

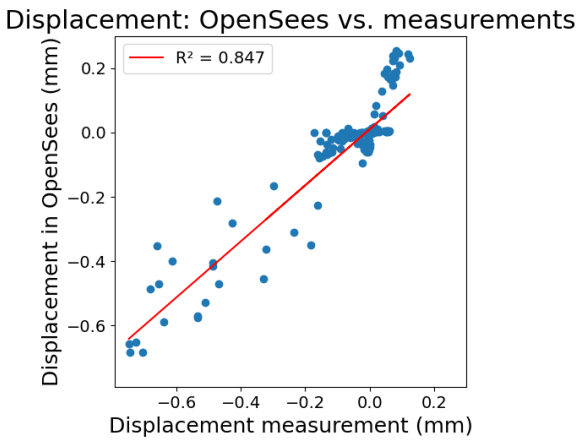


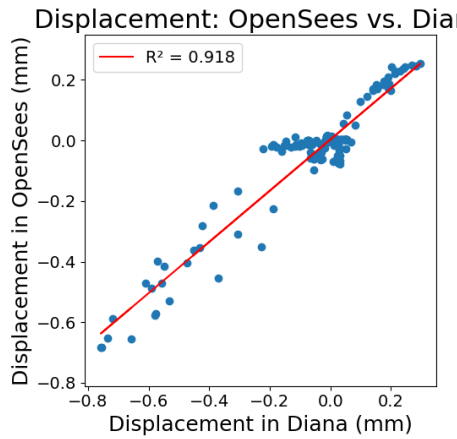
Validation: Displacement



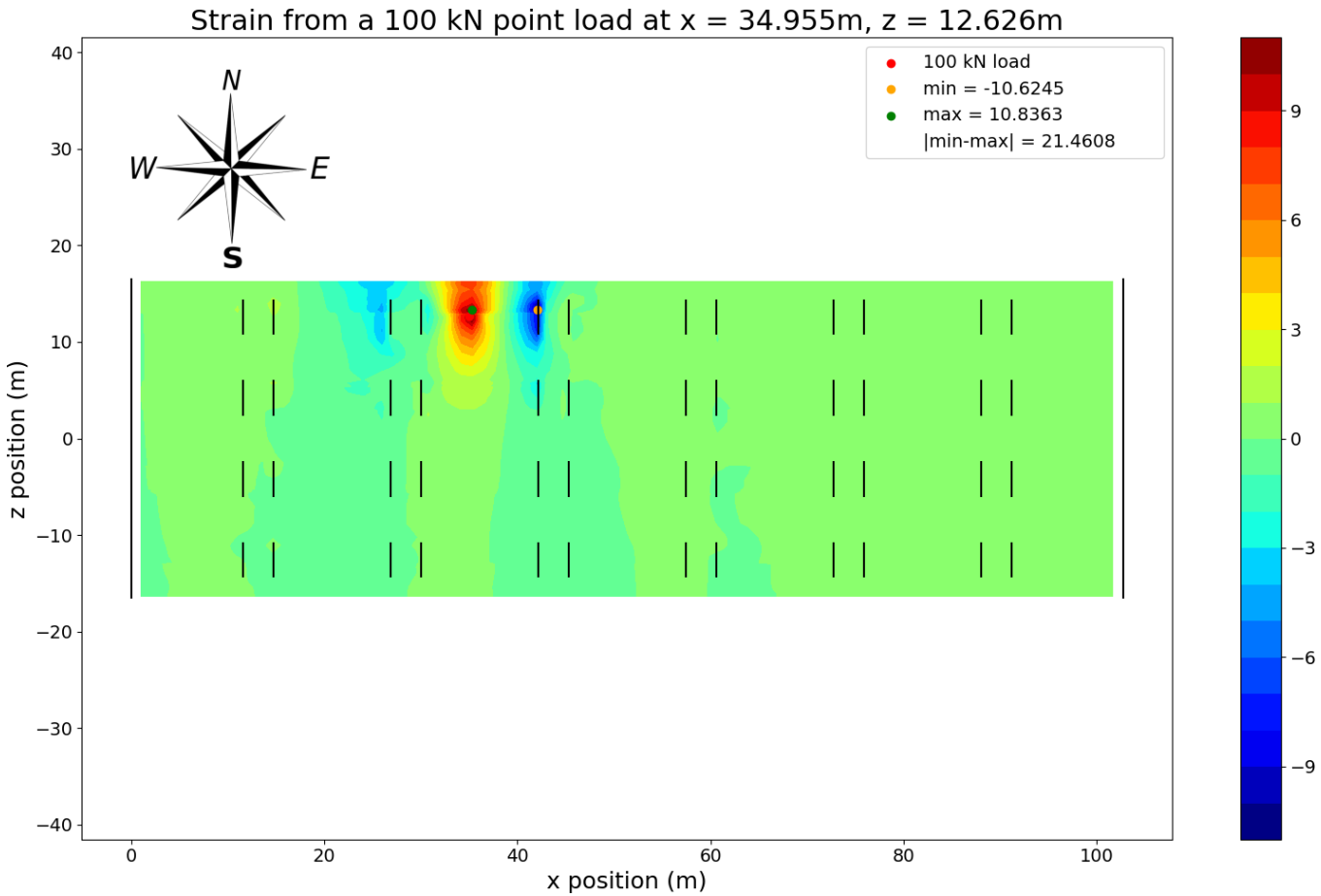
Validation: Displacement



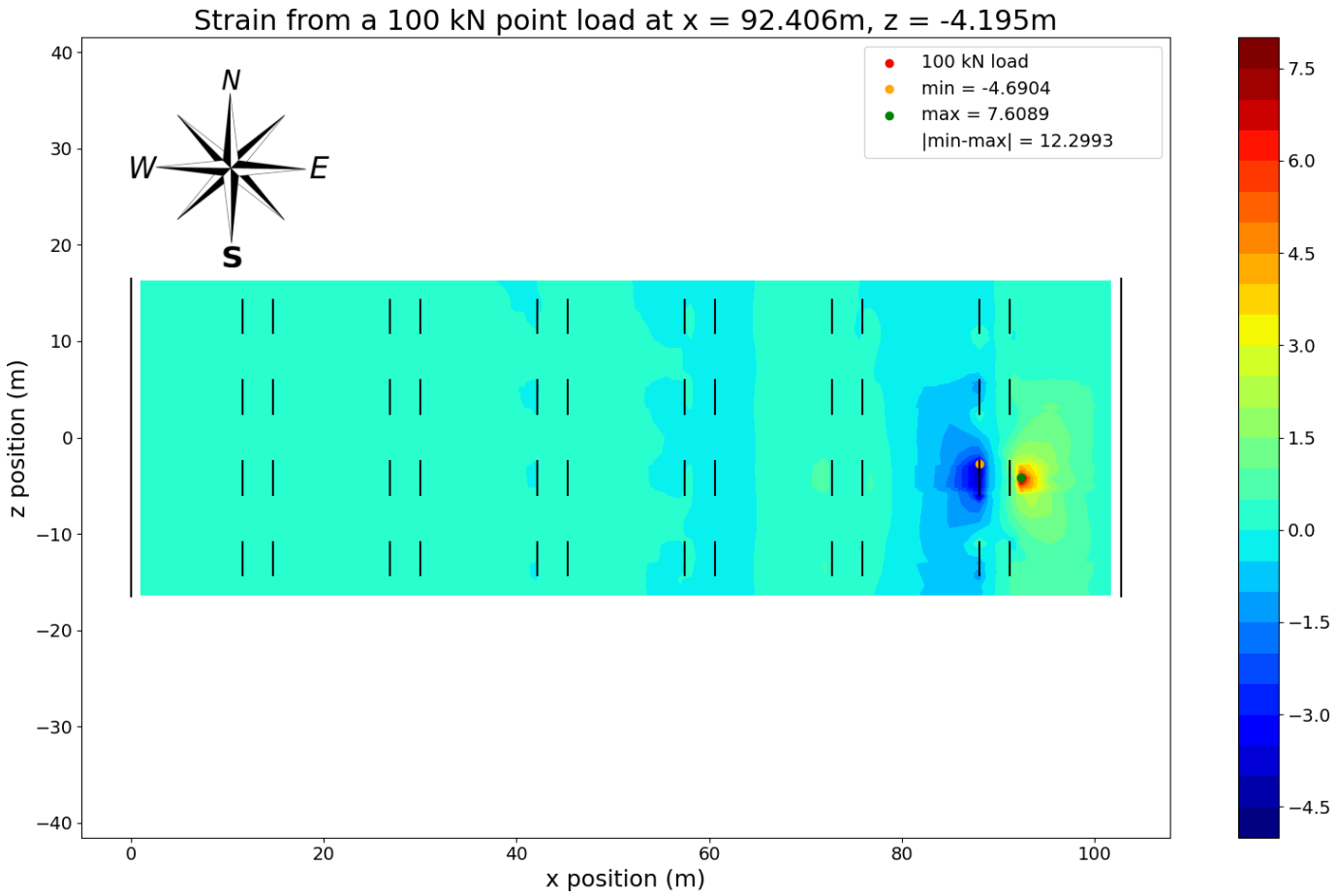




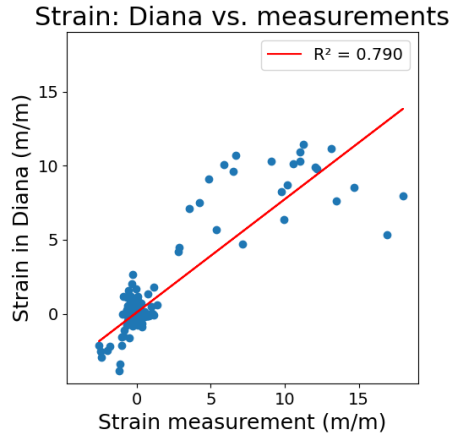
Validation Strain: OpenSees

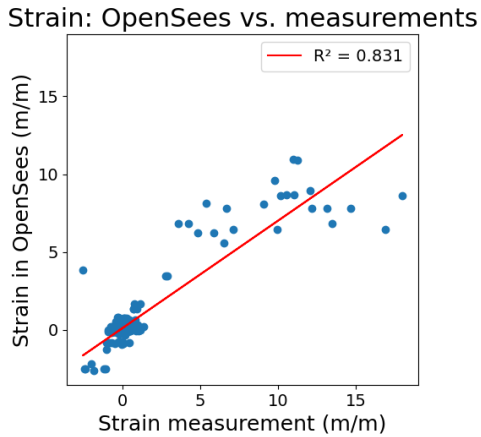


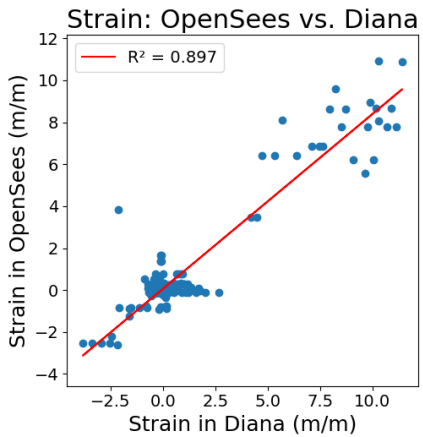
Validation Strain: OpenSees



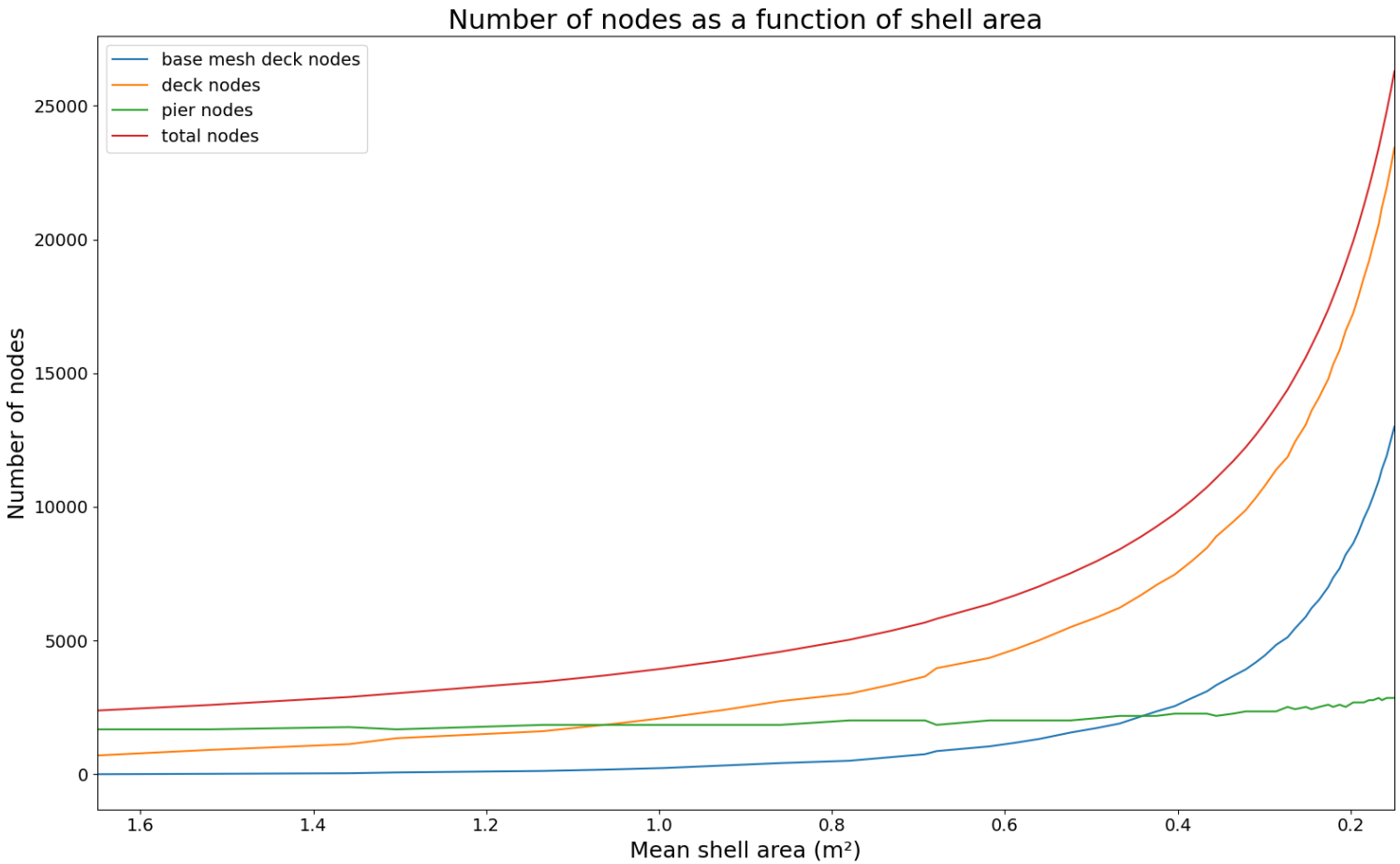
Validation: Strain



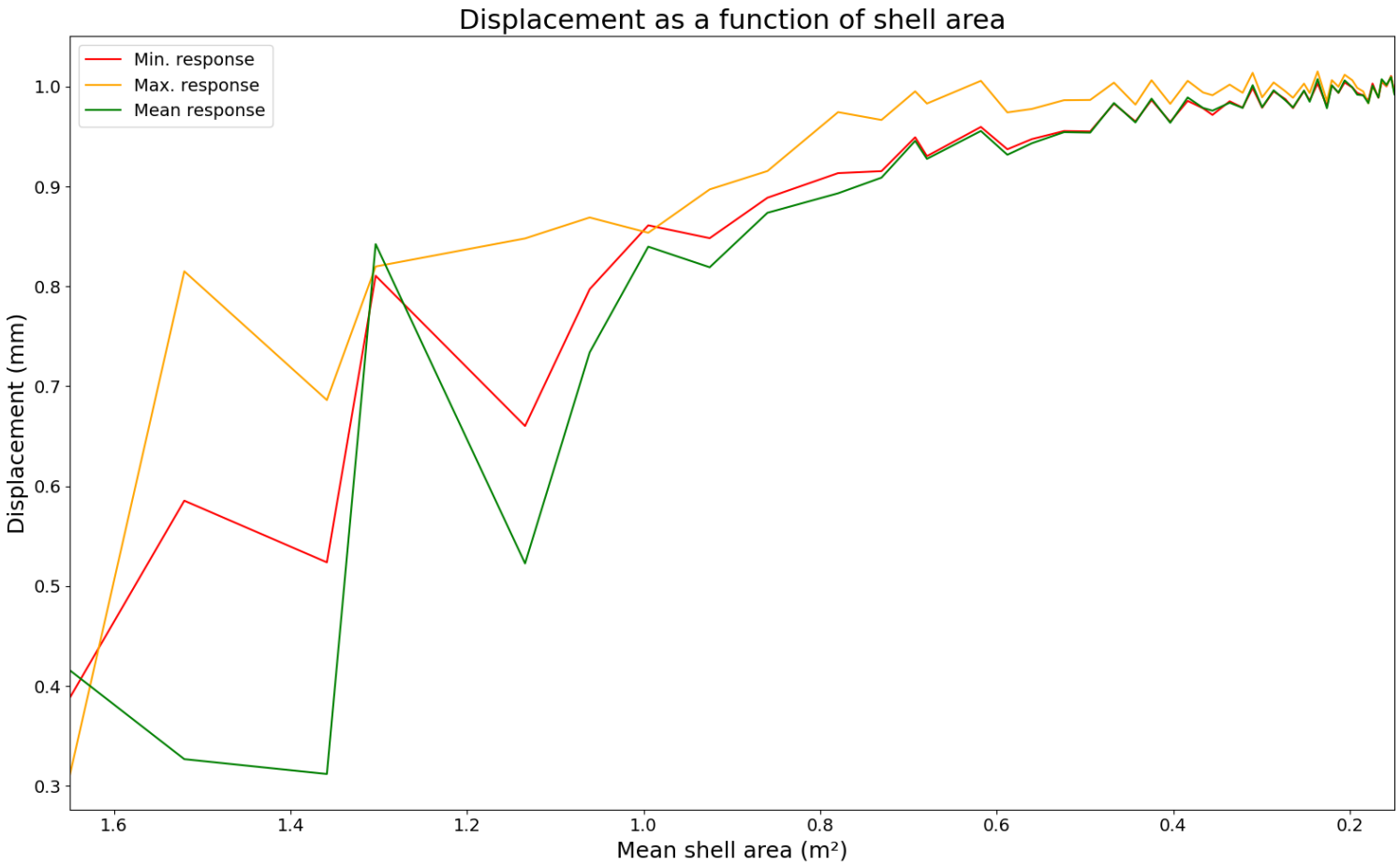




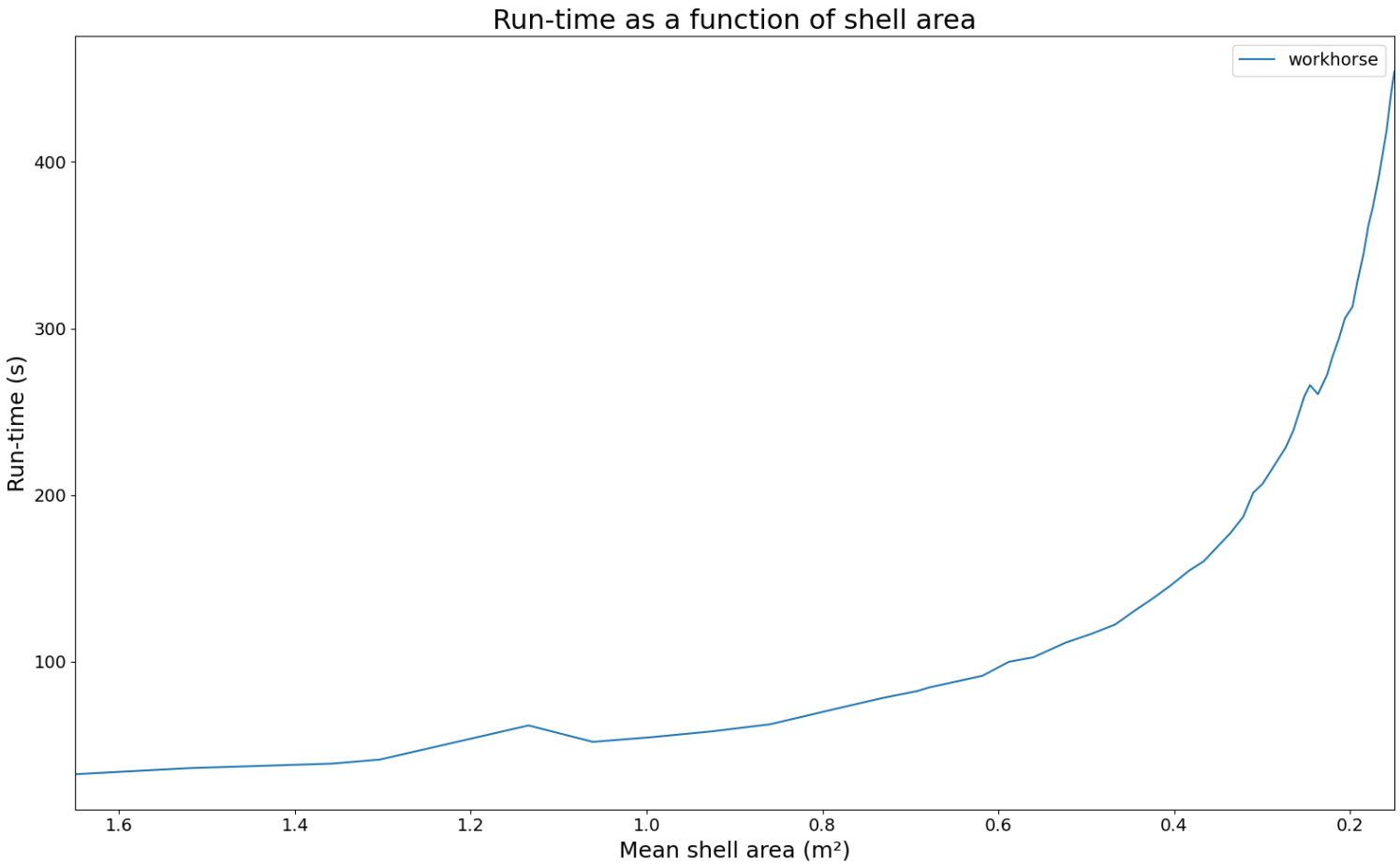
Model size


Number of nodes as a function of shell area
base mesh deck nodes
deck nodes
pier nodes
total nodes
Number of nodes
25000
20000
15000
10000
5000
0
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
Mean shell area (m²)

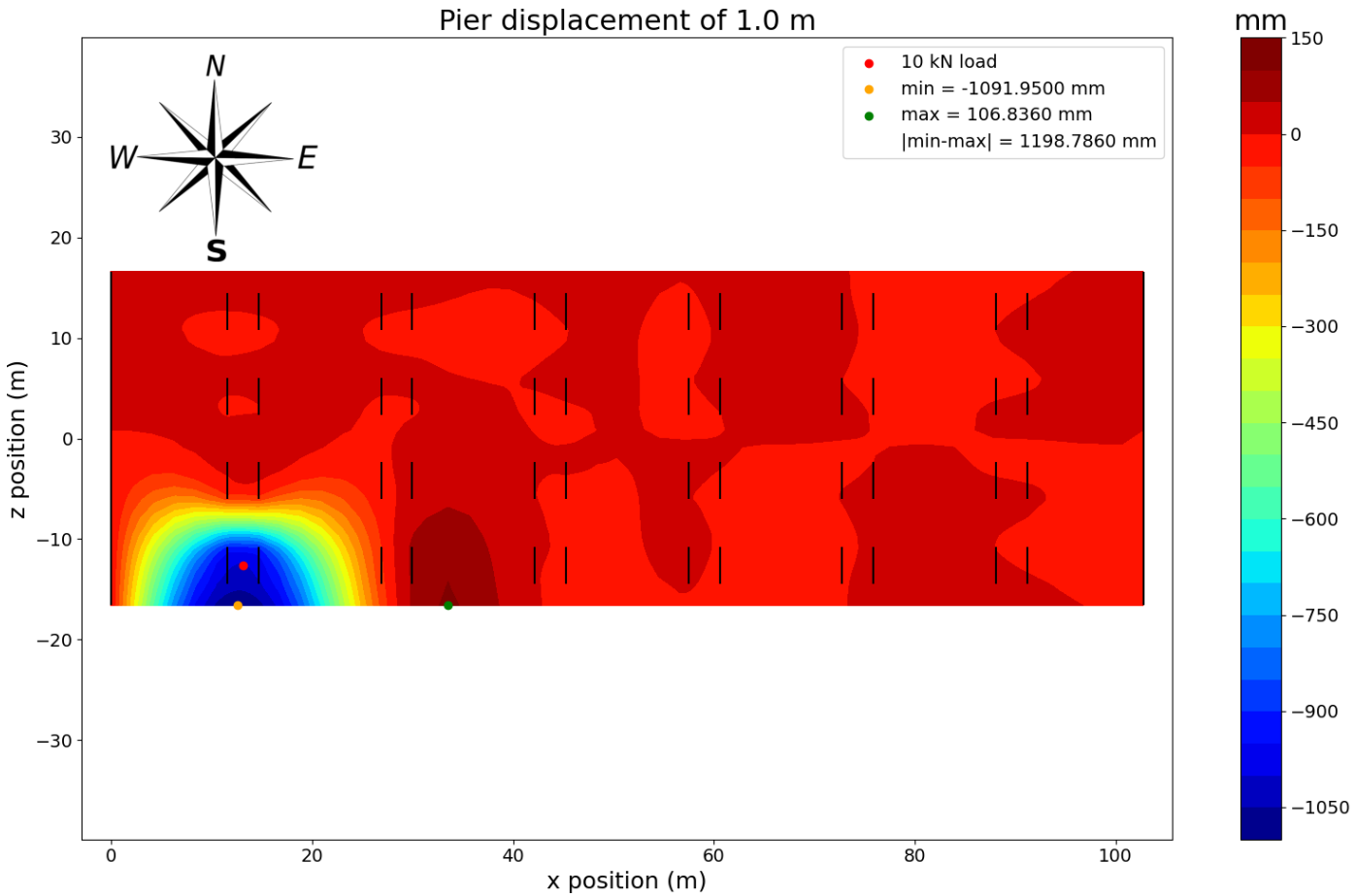
Model convergence



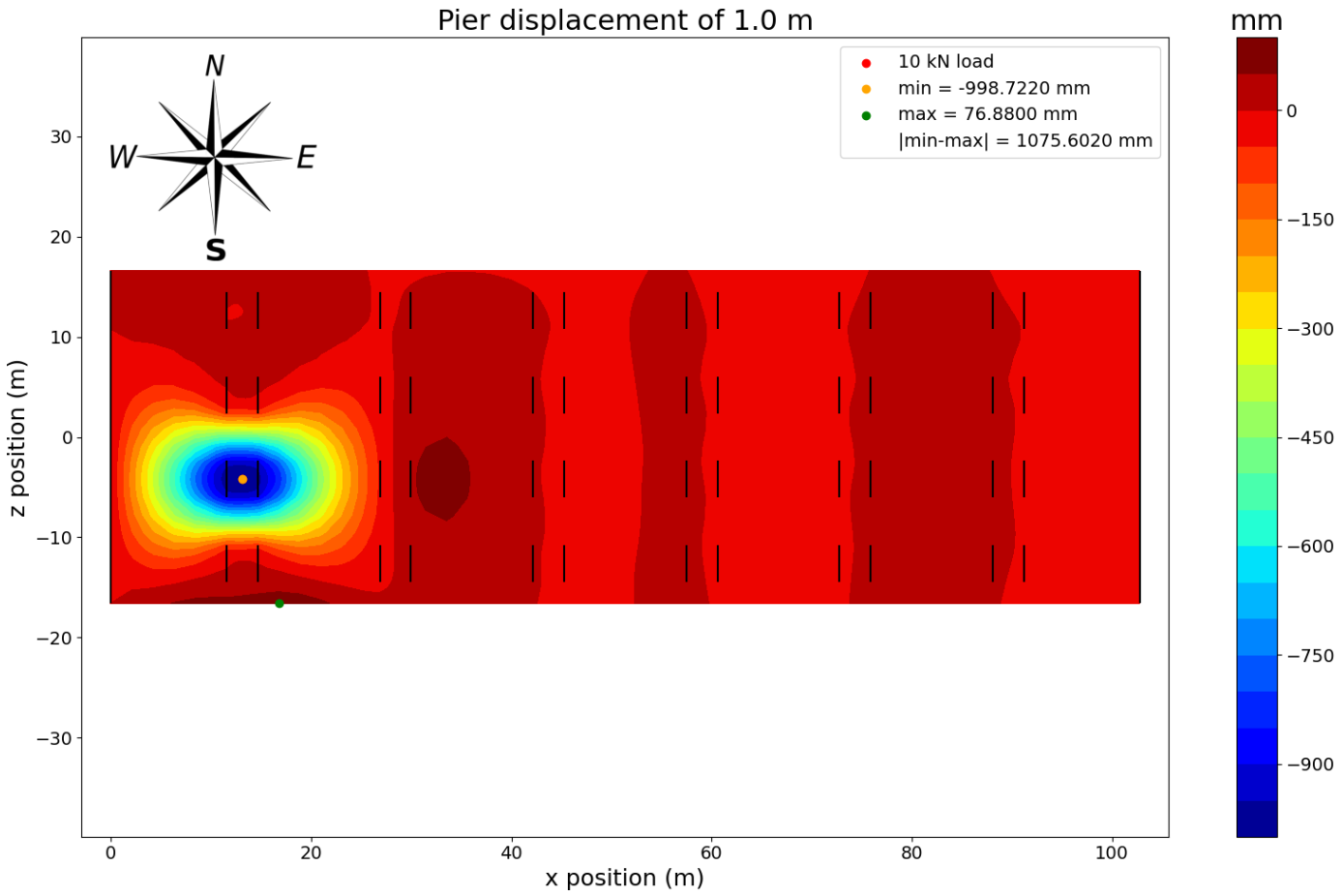
Model run-time



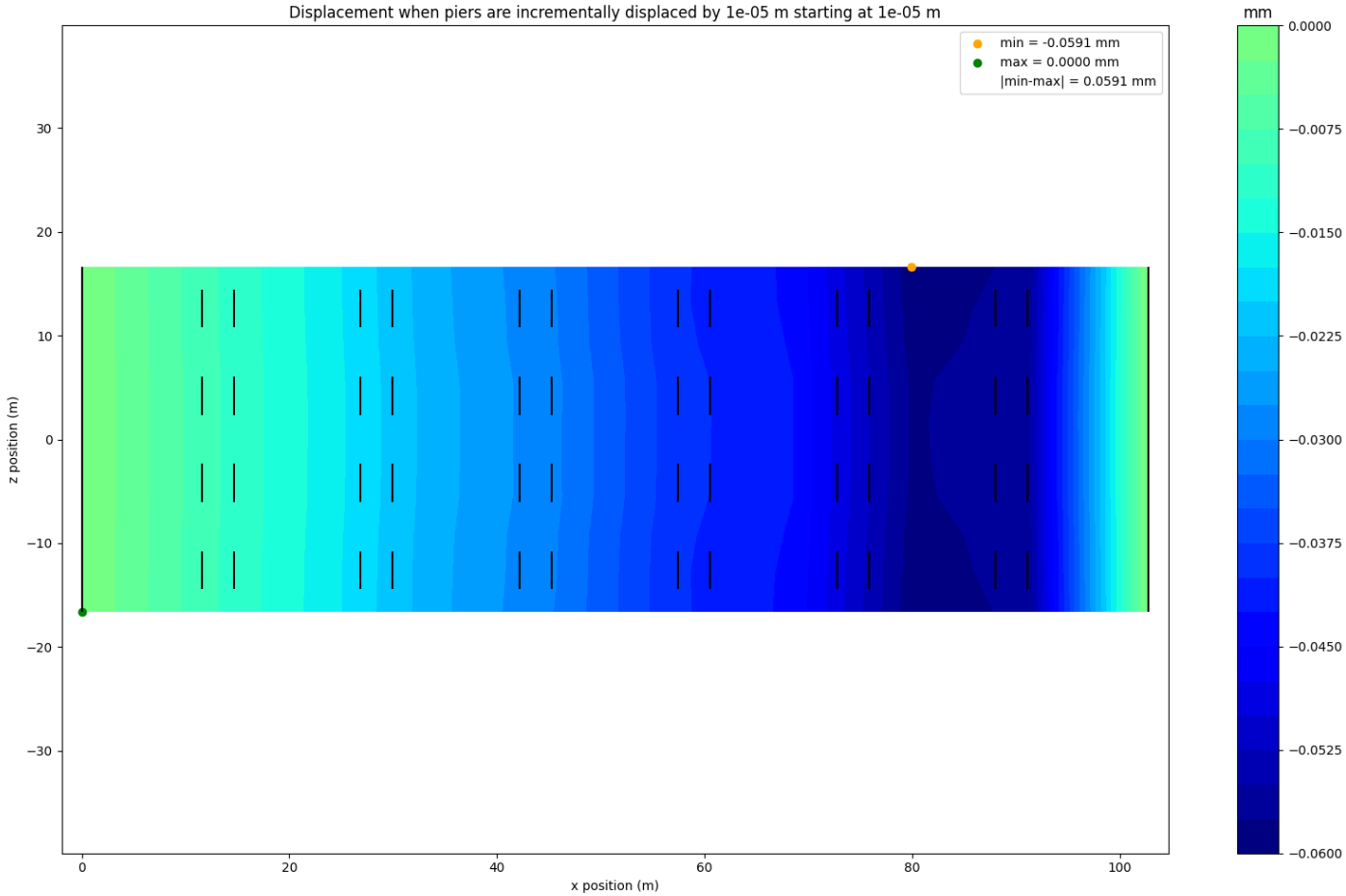
Pier Settlement



Pier Settlement

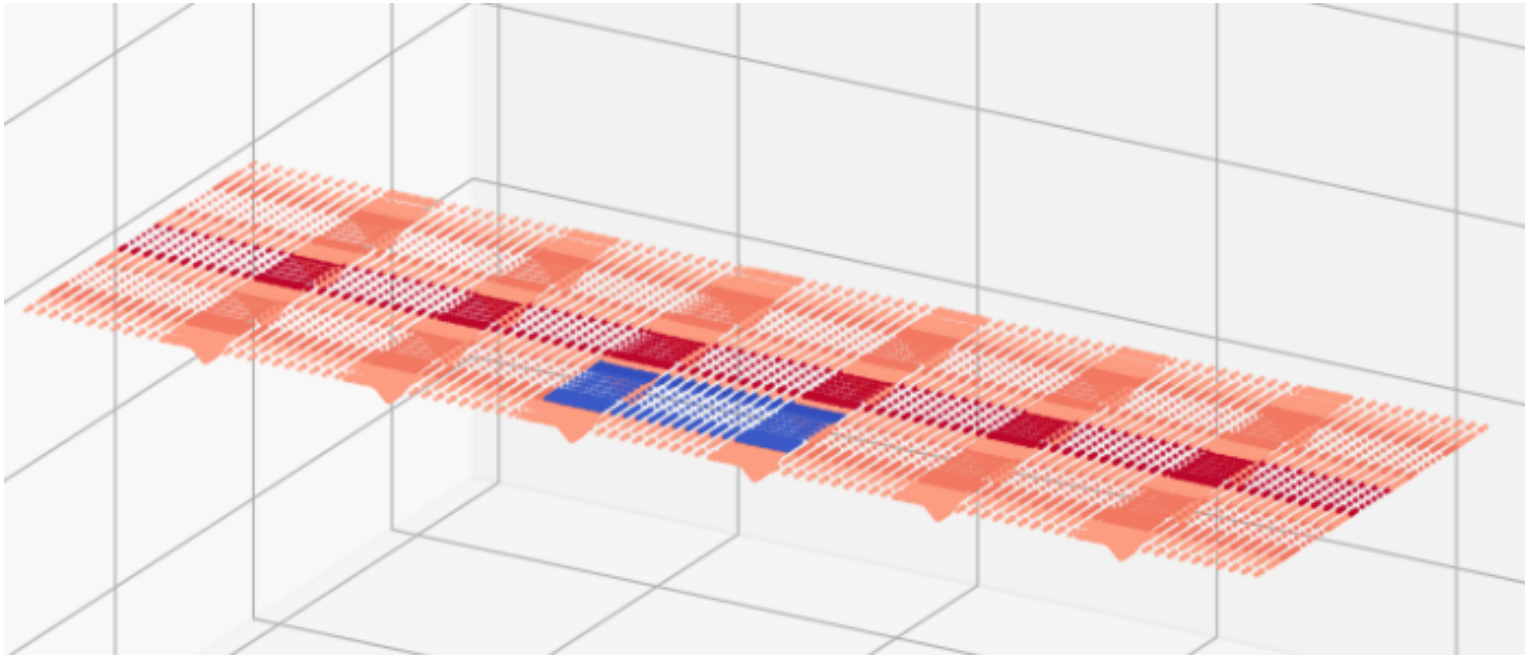


Pier Settlement

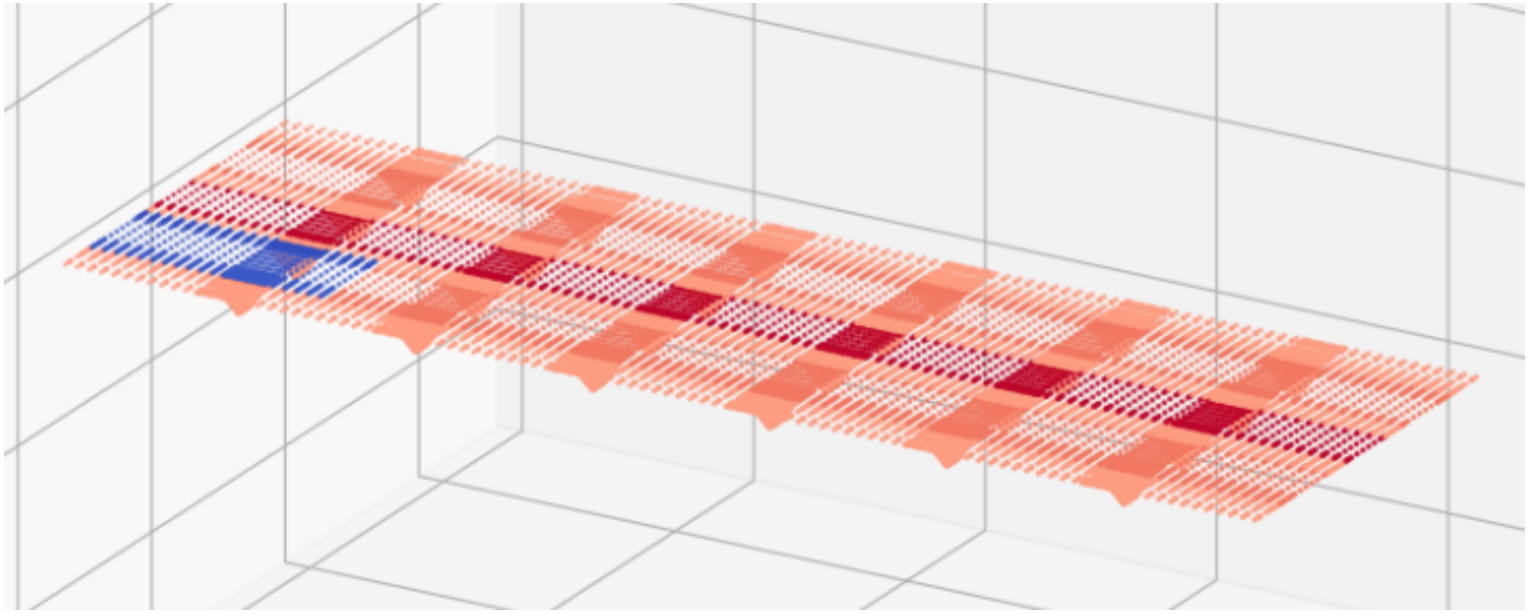


Crack Zone

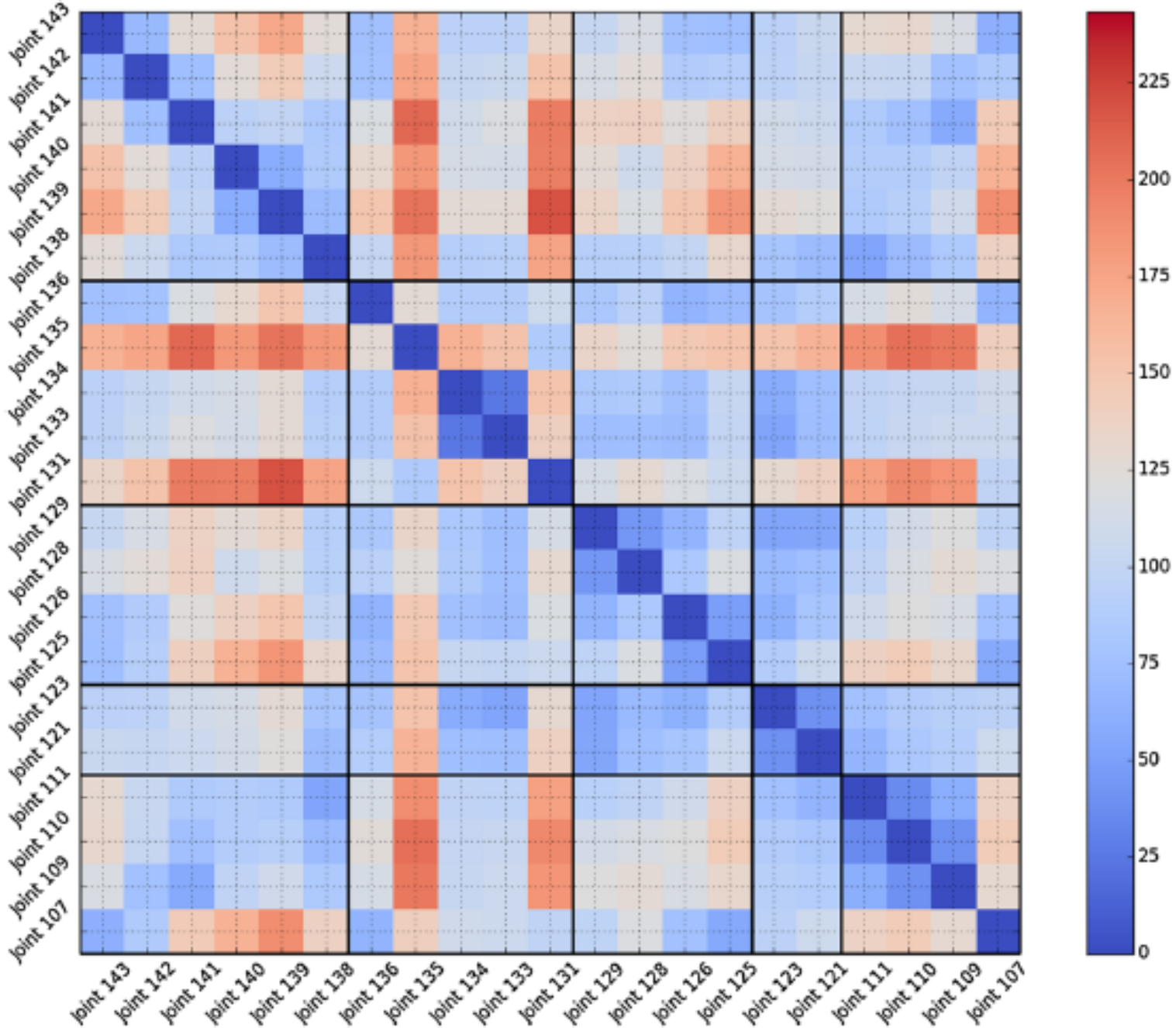
- Young's modulus *= 1/3



Crack Zone



Sydney Harbour Bridge

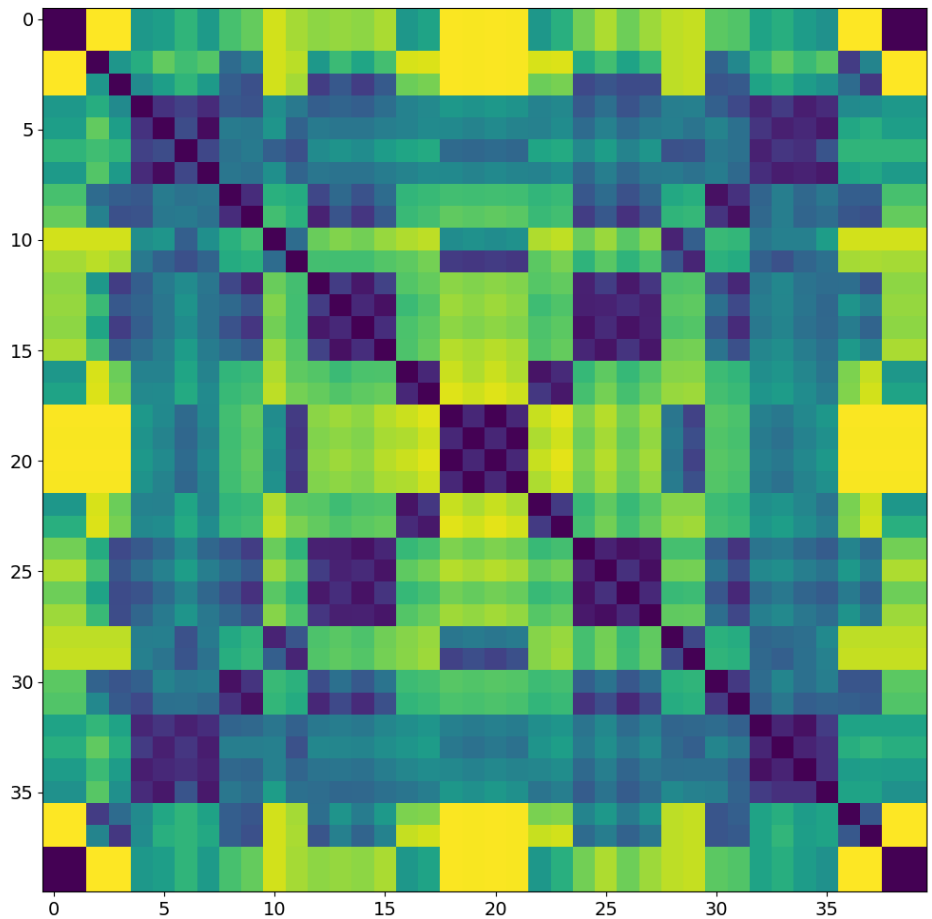


Baseline Detection

- "baseline" as in
  - compares to a previous baseline (snapshot)
  - baseline method to compare methods to

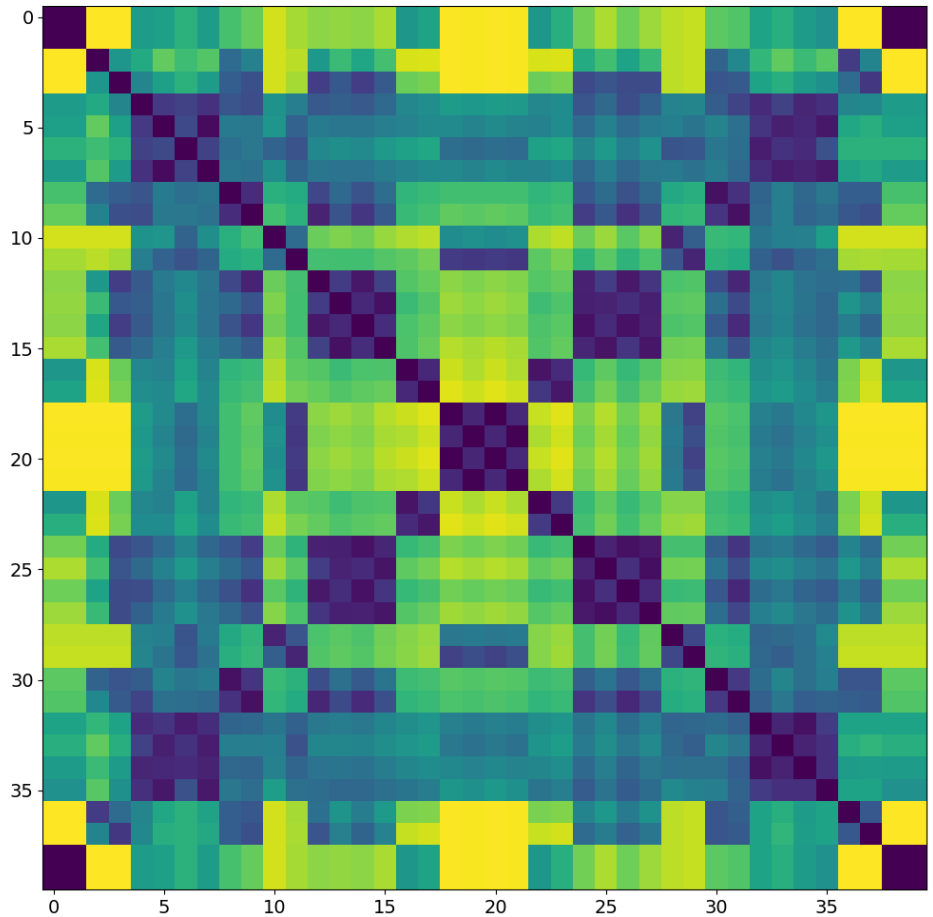
Baseline Detection

- at a point in time, record snapshot
- compare each sensor to each sensor: `C0_pi_pj`



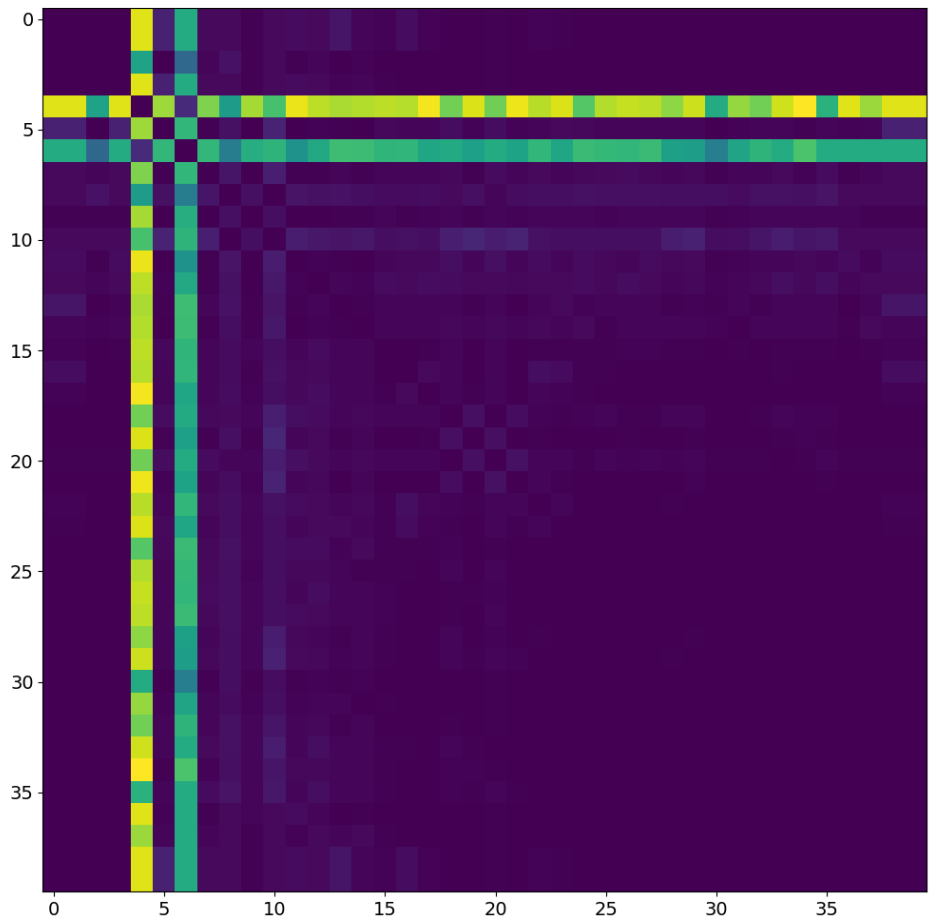
Baseline Detection

- at a later time
- compare each sensor to each sensor: `C1_pi_pj`



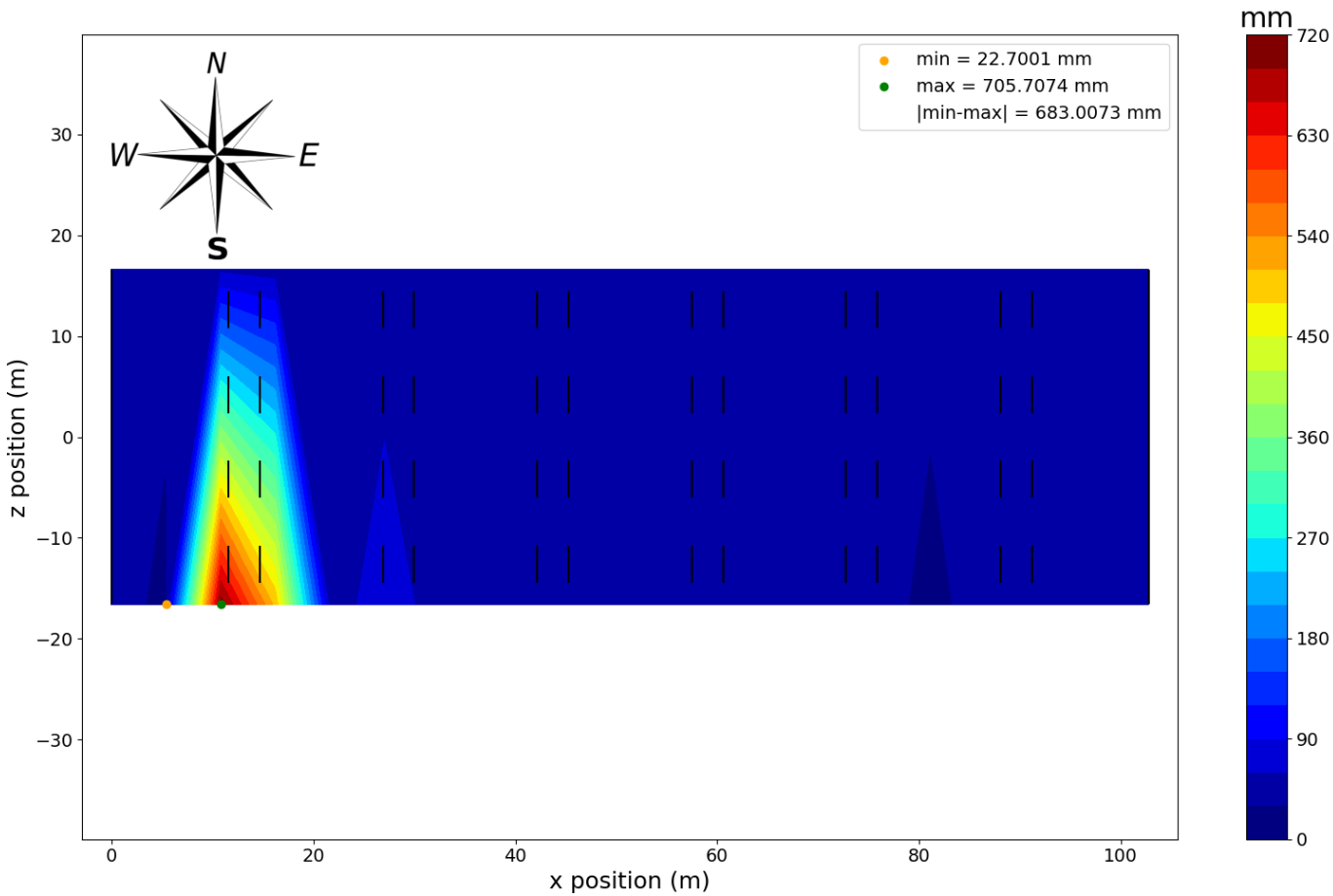
Baseline Detection

- determine difference between snapshots
- `D_pi_pj = | C0_pi_pj - C1_pi_pj |`



Baseline Detection

- map `D_pi_pj` to sensor positions



December - February

- December Remainder
  - Model agreement
  - Temperature in model
  - First ML-based damage experiment
- January/February
  - Further classification...

Questions for You

- Data on passenger vehicles?
  - Axles distances and weights
- Typical size and position of crack zones?
  - Concrete slab bridges
- Convinced by 5 variables for classification experiments?
  - vehicle loading
  - temperature load
  - pier settlement
  - cracked concrete
  - sensor noise
- What needs better explaining in this presentation?