Master Thesis

Building Technology
Graduation Studio

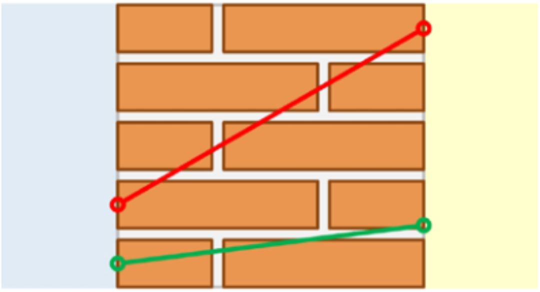
P5 presentation
January 13, 2025

Rik Trip
5421195

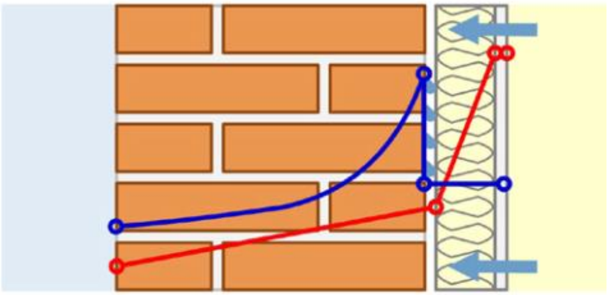
The hygrothermal performance of embedded wooden beams in solid brick masonry with interior capillary active insulation

Problem statement

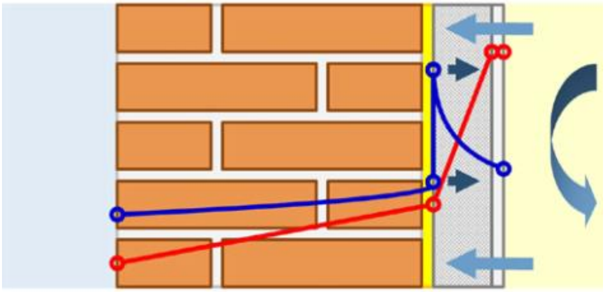
- Need to reduce the use of finite materials and mitigate climate change by reducing greenhouse gas emissions and building energy consumption
- The application of interior insulation can cause hygrothermal risks due to changes in hygrothermal performance of the wall
- More insight into the effect of these changes is needed and possible solutions, like capillary active insulation, are further researched



Without insulation



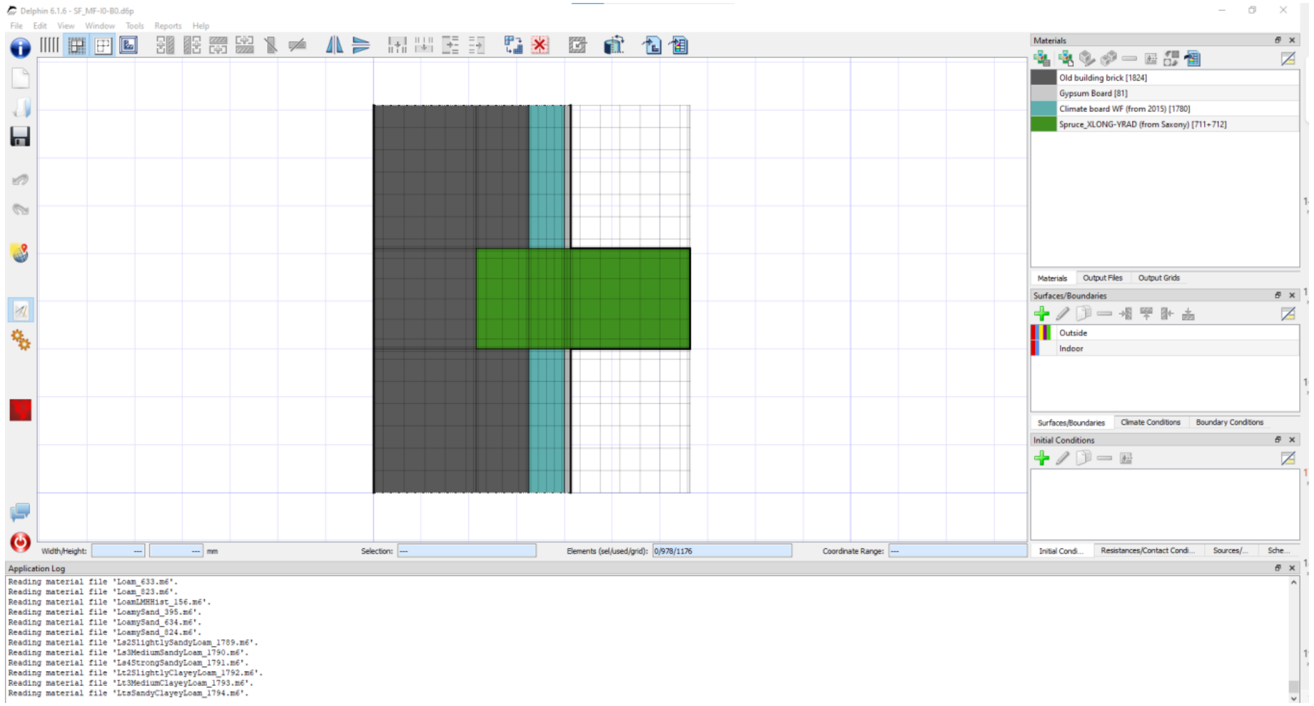
With standard insulation



With cap. act. insulation

Problem statement

- Hygrothermal simulation software to simulate performance in a relative short amount of time



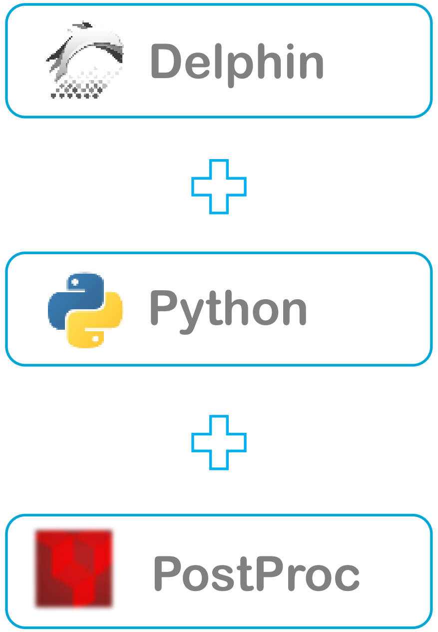
Objective

To provide insight into the key parameters effecting the hygrothermal safety of embedded wooden beams in solid brick masonry when interior capillary active insulation is applied

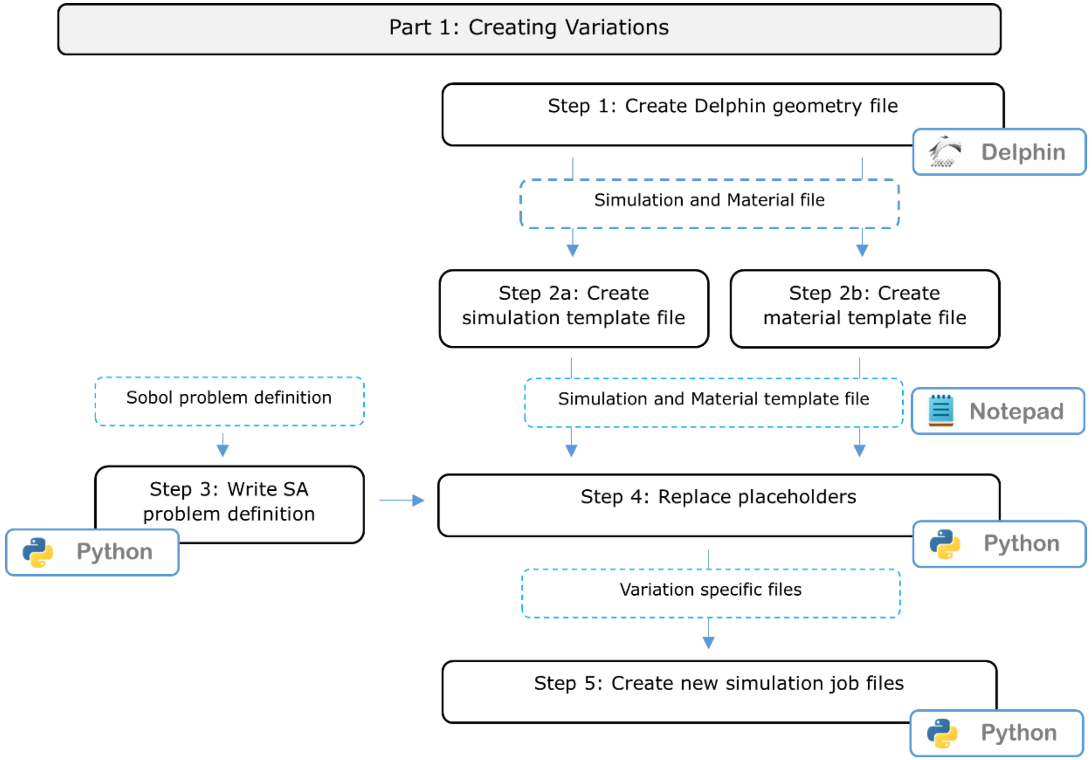
Objective

- What are the key parameters that affect the hygrothermal simulation most?
  - Sobol sensitivity analysis
  - Delphin hygrothermal simulation software
  - Python programming

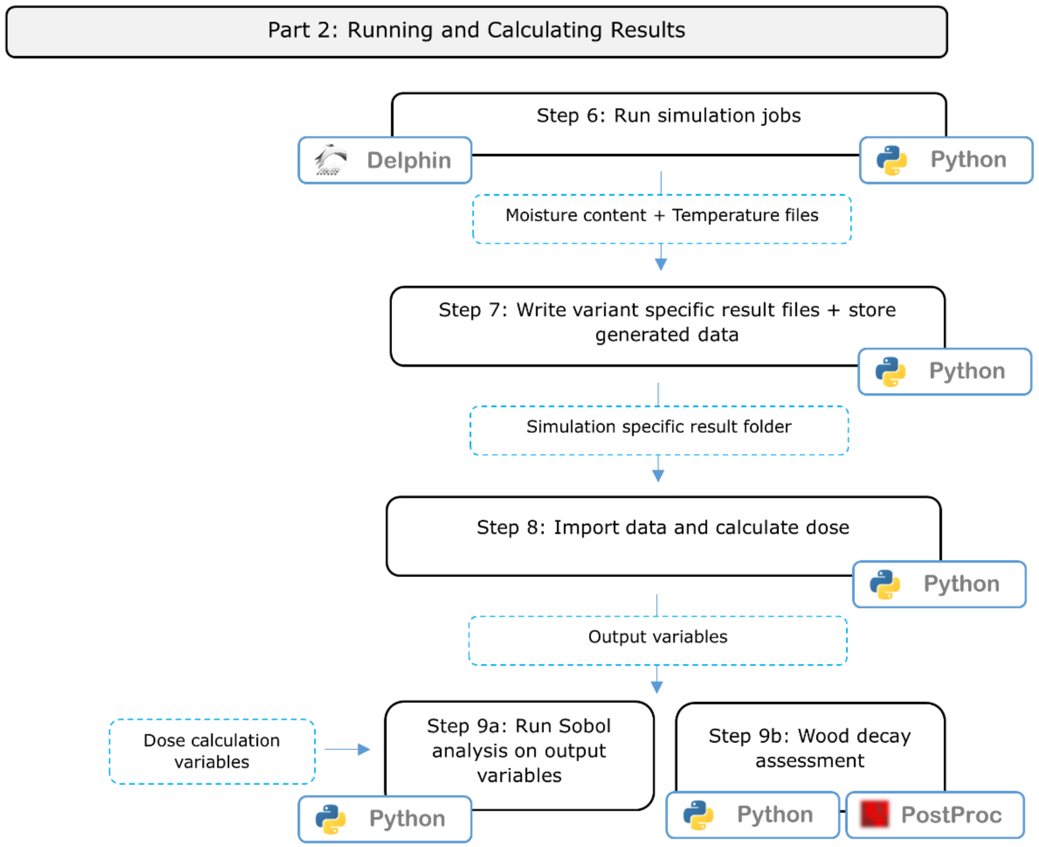
- How can the hygrothermal performance of a building component be visualised?
  - Delphin hygrothermal simulation software
  - Python programming
  - Delphin postprocessor PostProc



Methodology



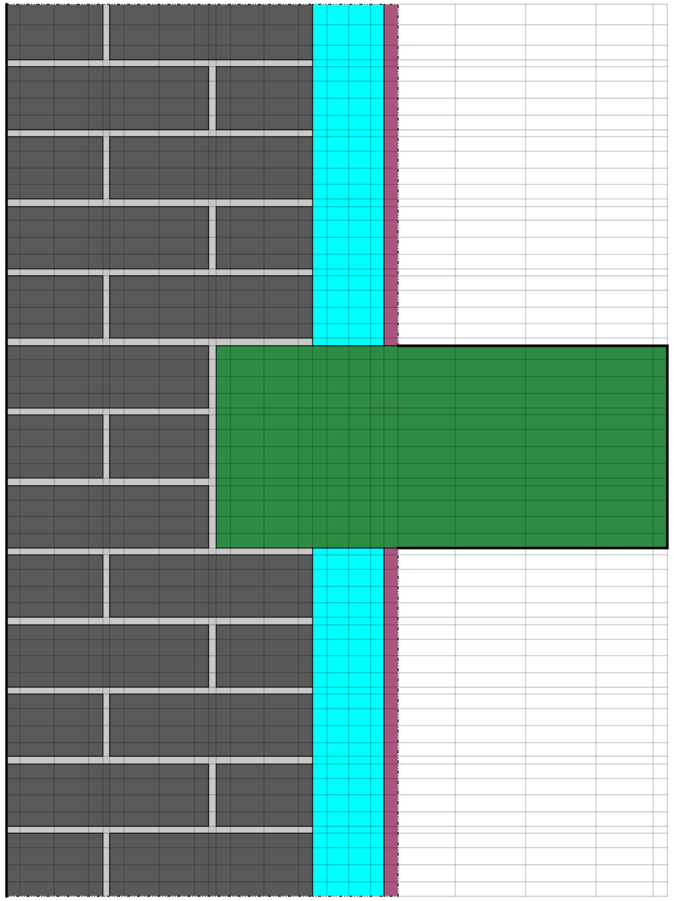
Methodology



Methodology - Geometry

- One-and-a-half solid masonry wall with 200mm high embedded wooden beam and 75mm of insulation + gypsum finishing board

- Location: Essen, Germany
  - Orientated: 270° (west) + 90° (vertical)
  - Outdoor dynamic measured weather data, including temperature, short/long wave solar radiation, relative humidity and rain load
  - Indoor dynamic WTA 6.2 + 5% increased moisture load



Methodology – Material properties

	unit	Gypsum [ID: 81]	Wood X/Y [ID: 711/712]	Brick [ID: 1824]	Insulation [ID: 1780]
Density	$[\mathrm{kg/m^3}]$	850	393.7	1698.3	186.9
Specific heat capacity	$[\mathrm{J/kgK}]$	850	1843	929	1100
Thermal conductivity	$[\mathrm{W/mK}]$	0,2	0.11/0.15	0.599	***VAR***
Vapour diffusion resistance factor	[-]	10	186.1/4.6	***VAR***	***VAR***
Open porosity	$[\mathrm{kg/m^3}]$	650.0	737.5	359.1	929.5
Effective saturation	$[\mathrm{kg/m^3}]$	551.0	728.1	324.1	928.8
Hygroscopic sorption value at RH= 80%	$[\mathrm{kg/m^3}]$	7.2	58.9	*CAL*	*CAL*
Water uptake coefficient	$[\mathrm{kg/m^3/s^{0,5}}]$	0.28	0.012	0.097	0.85
Pore size distribution factor (n-value)	[-]	-	-	***VAR***	***VAR***

Methodology – Sensitivity analysis

- Run Sobol sample generator with parameters, bounds, number of variations and output indices
- N-value represents the accuracy and reliability of the sensitivity analysis and determines the number of samples generated and is set to 128
- Parameter data frame saved to be used multiple times

```python
# *** SETUP ***
# 1. IMPORT
import pandas as pd
from SALib.sample import sobol

# *** MAIN ***
# 1. SETUP SENSITIVITY ANALYSIS PROBLEM DEFINITION
    # * Change *: Optionally change parameters and/or bounds of SA

problem = {
    'num_vars': 5,                                                 #AMOUNT OF PARAMETERS
    'names' : ['I_Thermal Conductivity',                           #I_LAMBDA
               'I_Water Vapor Diffusion Resistance Factor',        #I_MEW
               'I_NValue',                                         #NEW_STORAGE_FUNCTIONS:
                                                                       #I_THETA_CAP
                                                                       #I_THETA_80
               'B_Vapor Diffusion Resistance Factor',              #B_MEW
               'B_NValue'],                                        #NEW_STORAGE_FUNCTIONS:
                                                                       #B_THETA_CAP
                                                                       #B_THETA_80

    'bounds' : [[0.01875, 0.075],                                  #I_LAMBDA
                [1, 100],                                          #I_MEW
                [1.5, 2.75],                                       #I_N-VALUE
                [5, 50],                                           #B_MEW
                [1.2, 2]]                                          #B_N-VALUE
}

# 2. CREATE SOBOL SAMPLE COLLECTION and CREATE CSV FILE
    # * Change *: n value of accuracy of sensitivity analysis
sobol_samples_values = sobol.sample(problem, N: 128, calc_second_order=True)
sobol_samples_for_csv = pd.DataFrame(sobol_samples_values)
print(sobol_samples_for_csv)
    # * Change *: Path from content root to dataframe csv storage location
sobol_samples_for_csv.to_csv( path_or_buf: 'SA_Variants/SA_Sobol Samples v3.csv', index=False)
print(r'Sobol sample dataframe is created and saved to directory')
```

Methodology – Simulation variants

- Three template files with placeholders are used to generated simulation jobs: {MF-I01.m6}, {MF-B01.m6}, {SF_MF-I01-B01.d6p}

- Data file layout for 1 simulation:
  - Moisture content of simulation 1:
    - [Day 1: [Coordinate 1, Coordinate 2, Cn]
    - Day 2: [Coordinate 1, Coordinate 2, Cn]
    - Day 2: [Coordinate 1, Coordinate 2, Cn]]
  - Temperature content of simulation 1:
    - [Day 1: [Coordinate 1, Coordinate 2, Cn]
    - Day 2: [Coordinate 1, Coordinate 2, Cn]
    - Day 2: [Coordinate 1, Coordinate 2, Cn]]

```python
# 3. LOOP THROUGH MATERIAL VARIANTS
for i in range(1536): #add later instead of 1: len(sobol_samples) (or: 1536)
    mc_list_brick = []
    mc_list_insulation = []

    # 3A. IMPORT FROM SOBOL DATAFRAME
    i_lambda = sobol_samples.iloc[i, 0]
    i_mew = sobol_samples.iloc[i, 1]
    i_nvalue = sobol_samples.iloc[i, 2]
    b_mew = sobol_samples.iloc[i, 3]
    b_nvalue = sobol_samples.iloc[i, 4]

    # 3B. CALCULATE STORAGE FUNCTIONS
    pa_interval = np.logspace( start: 0,  stop: 12,  num: 50)
    #print(pa_interval)
    pa_interval_log = np.log10(pa_interval)

    for j in range(len(pa_interval)):
        moisture_insulation = float(van_genuchten_theta(pa_interval[j], i_theta_r, i_theta_s, i_alpha, i_nvalue))
        mc_list_insulation.append(moisture_insulation)

    i_mc_thpc_x = " ".join([str(x) for x in pa_interval_log])
    i_mc_thpc_y = " ".join([str(x) for x in mc_list_insulation])
    i_mc_pcth_x = " ".join([str(x) for x in mc_list_insulation[::-1]])
    i_mc_pcth_y = " ".join([str(x) for x in pa_interval_log[::-1]])

    for k in range(len(pa_interval)):
        moisture_brick = float(van_genuchten_theta(pa_interval[k], b_theta_r, b_theta_s, b_alpha, b_nvalue))
        mc_list_brick.append(moisture_brick)

    b_mc_thpc_x = " ".join([str(x) for x in pa_interval_log])
    b_mc_thpc_y = " ".join([str(x) for x in mc_list_brick])
    b_mc_pcth_x = " ".join([str(x) for x in mc_list_brick[::-1]])
    b_mc_pcth_y = " ".join([str(x) for x in pa_interval_log[::-1]])
```

```python
# 4. CREATE LIST OF SIMULATION FILES
#create empty list with all simulation name and job files
list_simulation_files = []
jobs = []

#create simulation files for all different sobol variants
for j in range(1): #location options
    for l in range (1536): #material options (amount of sobol samples:1536)
        #replace placeholders in template with material and climate data and specify database directory
        #variant_2 = project_template.replace('${REP_LOCATION}', '{}'.format(climate_files[j][0]))
        variant_2 = project_template.replace( __old: '${REP_MATERIAL_INSULATION}', '{}'.format(list_material_files_insulation[l]))
        variant_2 = variant_2.replace( __old: '${REP_MATERIAL_BRICK}', '{}'.format(list_material_files_brick[l]))
        variant_2 = variant_2.replace( __old: '${REP_MATERIAL_DATABASE}',  __new: r'C:\Users\r.trip\OneDrive - Oosterhoff Group\Documents
        #variant_2 = variant_2.replace('${REP_CLIMATE_DATABASE}', r'C:\Users\r.trip\OneDrive - Oosterhoff Group\Documents\Graduation_

        #write simulation name and add to list
        simulation_file = "SF_MF-I{}-B{}.d6p".format( *args: l, l)
        write_file(VARIANTS_SUBDIR + "/Simulation/" + simulation_file, variant_2)
        list_simulation_files.append(simulation_file[:-4])

        #write simulation job and add to list
        if not os.path.exists(simulation_file[:-4] + "/var/restart.bin"):
            jobs.append([DELPHIN_EXECUTABLE, '-x', '--verbosity-level=0', '-p=2', 'SA_Variants/Simulation again/'+ simulation_file])


print(list_simulation_files)
print(f"Creation of simulation files done")

df_job_list = pd.DataFrame(jobs)
df_job_list.to_csv( path_or_buf: 'SA_Variants/SA_job files 2.csv', index = False)

df_list_simulation_files = pd.DataFrame(list_simulation_files)
df_list_simulation_files.to_csv( path_or_buf: 'SA_Variants/SA_simulation_files.csv', index = False)
```

Methodology – Dose calculation

- With the logistic dose-response model the dose is calculated using a moisture and temperature component
- Gives an indication of wood decay. Where the dose is a number in days

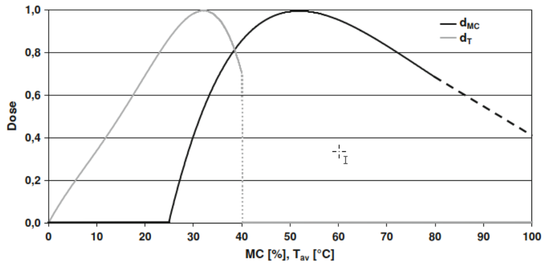
```python
# 3. CYCLE THROUGH INDEXES FOR ALL DAYS
#WITHIN EVERY SIMULATION FILE WE CYCLE THROUGH ALL THE DAYS (EVERY DAY SPECIFIC LIST OF THE SIMULATION LIST)
for day in range(aod): #LATER: PLACE 'AOD'

    moisture_content_at_day = moisture_content[day]
    temperature_at_day = temperature[day]

    #print(f"At day {day} the amount of coordinates = {len(moisture_content_at_day)}")

    #CHECK IF MOISTURE AND TEMPERATURE FILES HAVE THE SAME SIZE (AND THUS AMOUNT OF COORDINATES)
    if len(temperature) != len(moisture_content):
        print("Error: Temperature and moisture content lists must have the same length")

    #SAME LENGTH: CYCLE THROUGH ALL COORDINATES TO CALCULATE DOSE AT EVERY SPECIFIC COORDINATE
    else:
        # 4. CYCLE THROUGH COORDINATES FOR ONE SPECIFIC DAY
        #CREATE EMPY DOSE LIST FOR SPECIFIC DAY
        dose_list = []

        #FIND AND PROCESS COORDINATE SPECIFIC MOISTURE AND TEMPERATURE DATA
        for j in range(aoc):
            Dt = temperature_at_day[j]
            Du = moisture_content_at_day[j] * 100
            #print(f"\nTemperature: {Dt}, Moisture content: {Du} at coördinate {j}")

            #CHECK IF VALUES ARE WITHIN THERE LIMIT AND CALCULATE DOSE + ADD TO DOSE LIST
            if Du >= 25 and 0 <= Dt <= 40:
                Du_result = evar * Du**5 - fvar * Du**4 + gvar * Du**3 - hvar * Du**2 + ivar * Du - jvar
                Dt_result = kvar * Dt**4 + lvar * Dt**3 - mvar * Dt**2 + nvar * Dt
                #print(f"\nMoisture: {Du}, m component: {Du_result}. Temperature: {Dt}, t component: {Dt_result} at day {day} for coordinate {j}")

                dose = (avar * float(Dt_result) + float(Du_result)) * (avar + 1)**-1
                dose_list.append(round(dose, 1))
                #print ("The temperature and moisture content values are within there limits. \n"
                #       "Calculated: Du_result= " + str(round(Du_result, 3)) + ", Dt_result= " + str(round(Dt_result, 3)) + ", Dose= " + str(round(dose, 3)))

            # add zero to list if values are not within there limit
            else:
                dose_list.append(0)
                # print ("Temperature or moisture content is outside there limit\n")
```

$$D_u(u)^* = \ e * u^5 - f * u^4 + g * u^3 - h * u^2 + i * u - j$$

$$D_T(T)^* = k * T^4 + l * T^3 - m * T^2 + n * T$$

if Du > 0 and Dt > 0

$$D = (a * D_T[T] + D_u[u]) * (a + 1)^{-1}$$



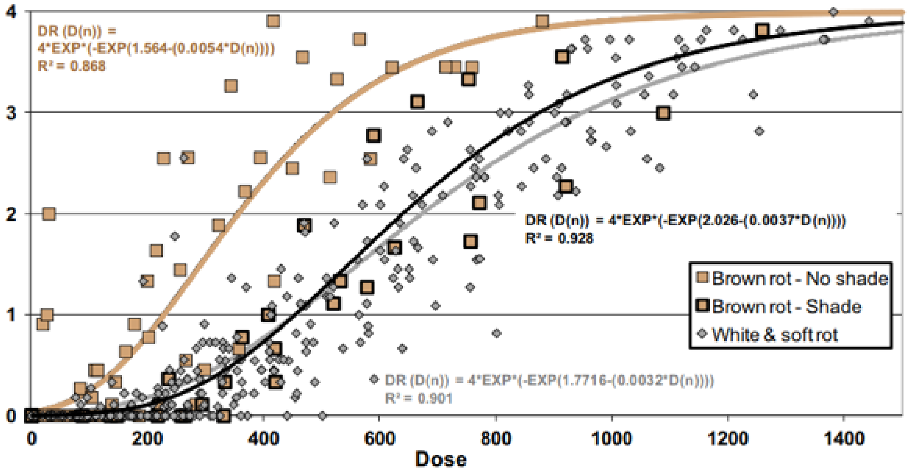
Methodology – Dose calculation

- Wood decay rating gives the progression of the decay

  - Decay rating 0 (no attack) -
  - Decay rating 1 (slight attack) 230 days
  - Decay rating 2 (moderate attack) 360 days
  - Decay rating 3 (severe attack) 520 days
  - Decay rating 4 (failure) 1400 days

- Inherent material resistance includes the effect of different wood species and corresponding wood properties

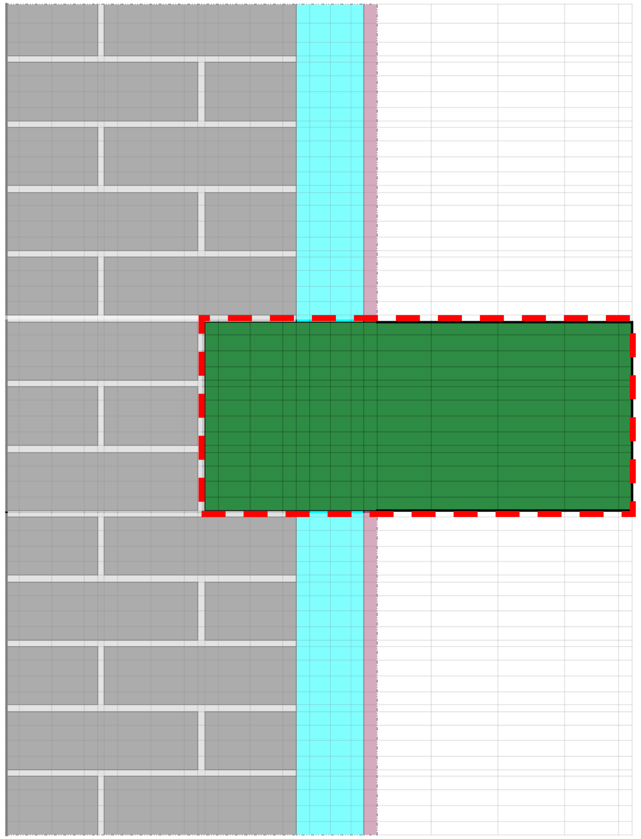
- Spruce as reference wood species



Methodology – Output variables

- Multiple analysis conducted on output variables
  - SA 1: Maximum moisture content in the beam
  - SA 2: Average moisture content in the beam

  - SA 3: Maximum dose value of the beam
  - SA 4: Average dose values of the beam



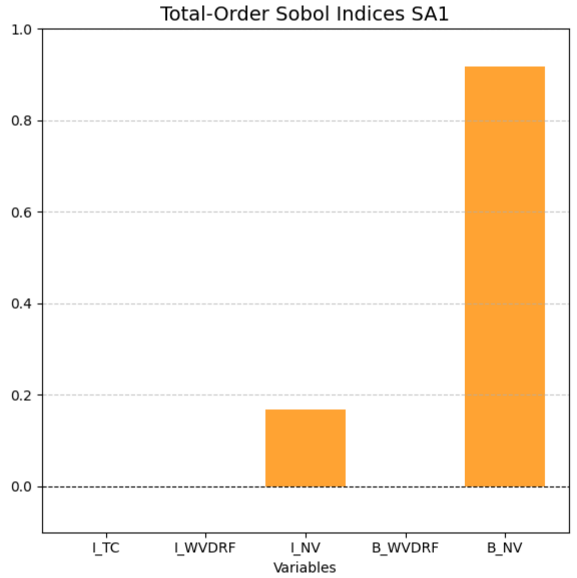
Geometry selection for sensitivity analysis

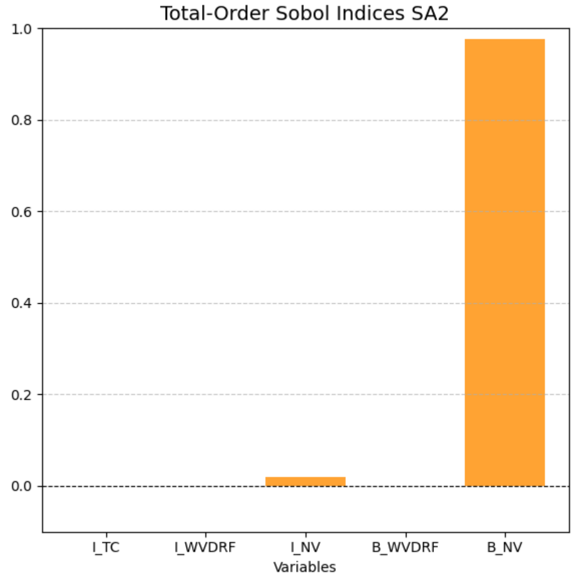
Results – Sensitivity analysis

- SA 1: Maximum moisture content in the beam
  - No influence of I_TC, I_WVDRF and B_WVDRF
  - 16.9% of I_NV and 91.8% of B_NV

- SA 2: Average moisture content in the beam
  - No influence of I_TC, I_WVDRF and B_WVDRF
  - 2 % of I_NV and 97% of B_NV

- Indicates I_NV is more influential in moisture extremes and B_NV in long term effects
- Uncertainty in results



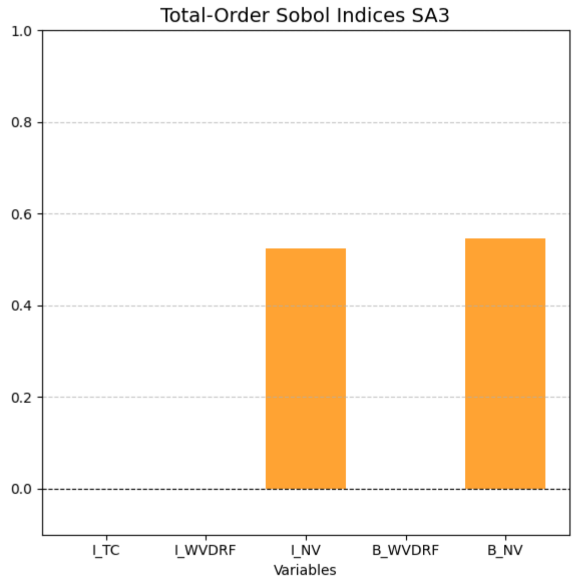


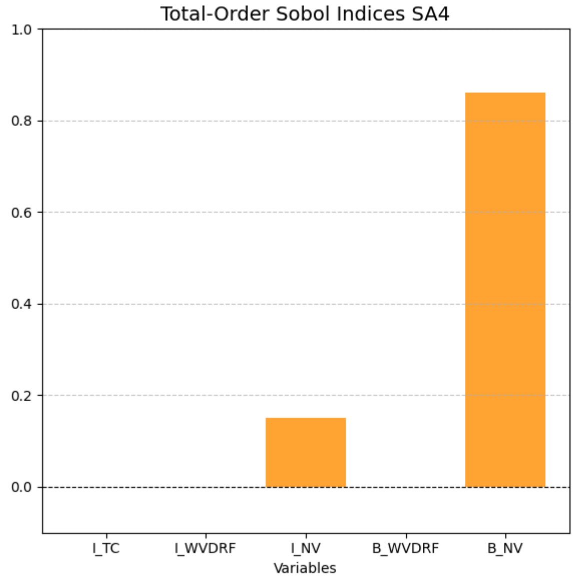
Results – Sensitivity analysis

- SA 3: Maximum dose content in the beam
  - No influence of I_TC, I_WVDRF and B_WVDRF
  - 52.4% of I_NV and 54.5% of B_NV

- SA 4: Average dose content in the beam
  - No influence of I_TC, I_WVDRF and B_WVDRF
  - 15.2% of I_NV and 86% of B_NV

- Indicates I_NV is more influential in moisture extremes and B_NV in long term effects
- Difference between maximum values as a result from dose boundary conditions

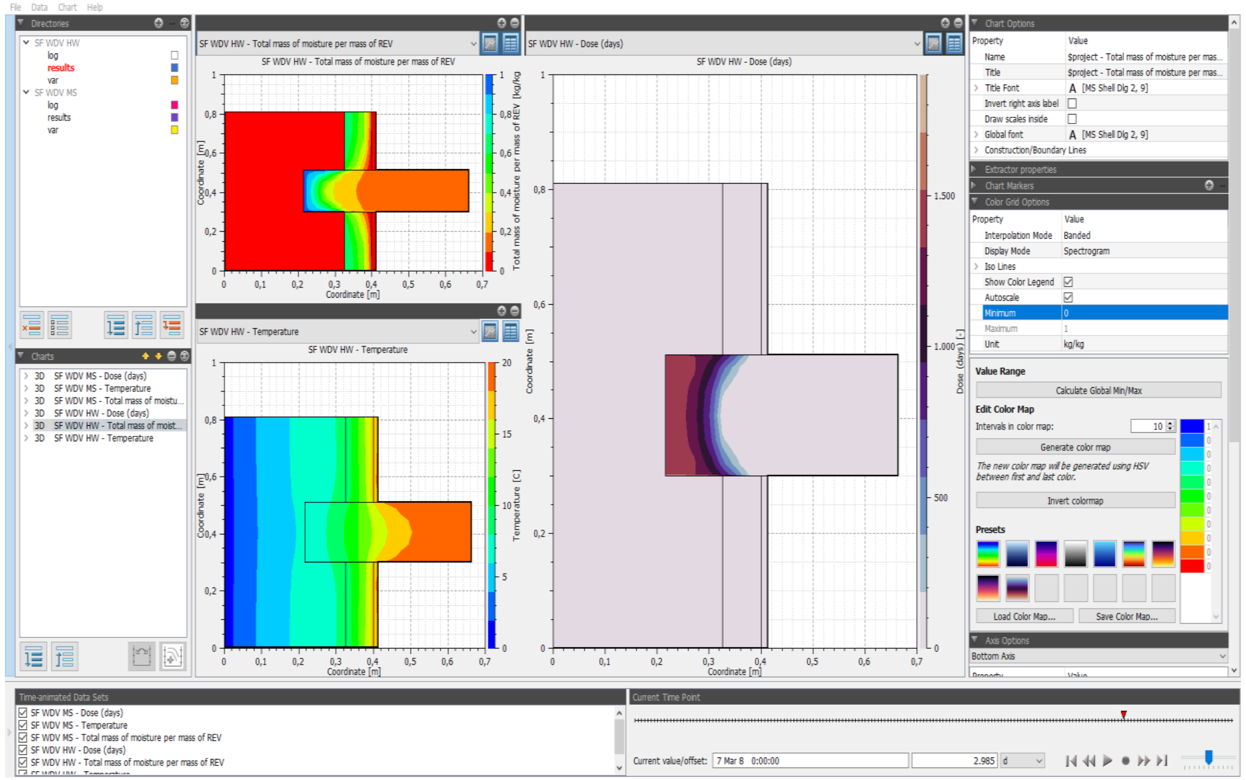




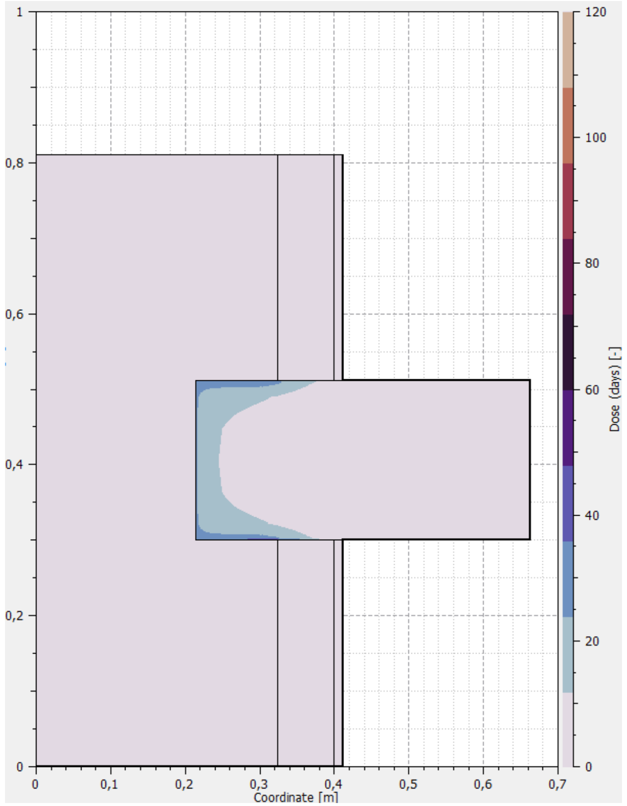
Results – Tool for wood decay assessment

- To visualise potential of wood decay
  - Development over time
  - Changes in boundary conditions
  - Changes in geometry
  - Changes in material type

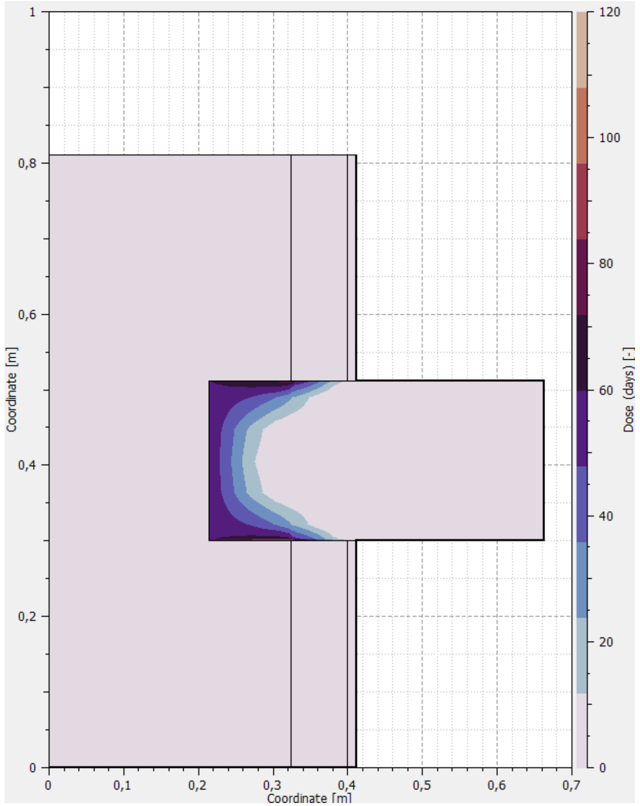
- Improve hygrothermal risk assessment



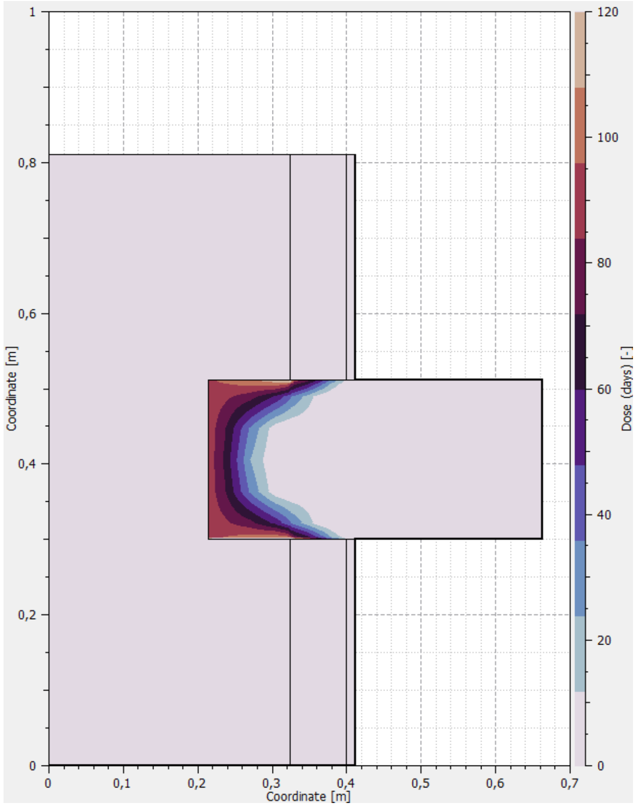
Results – Development over time



Dose after 1 year



Dose after 2 years



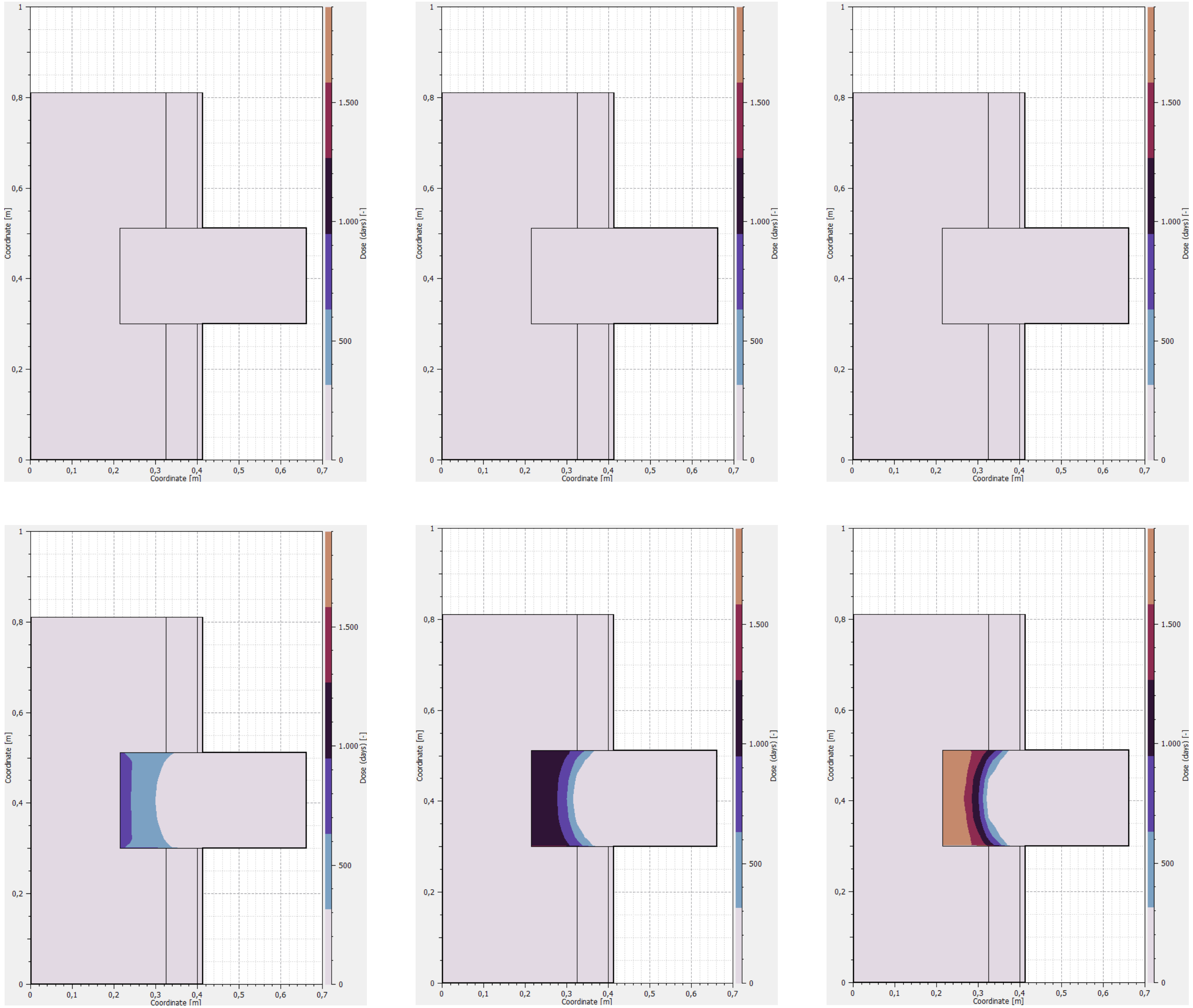
Dose after 3 years

Results

Wind driven rain

Without

With



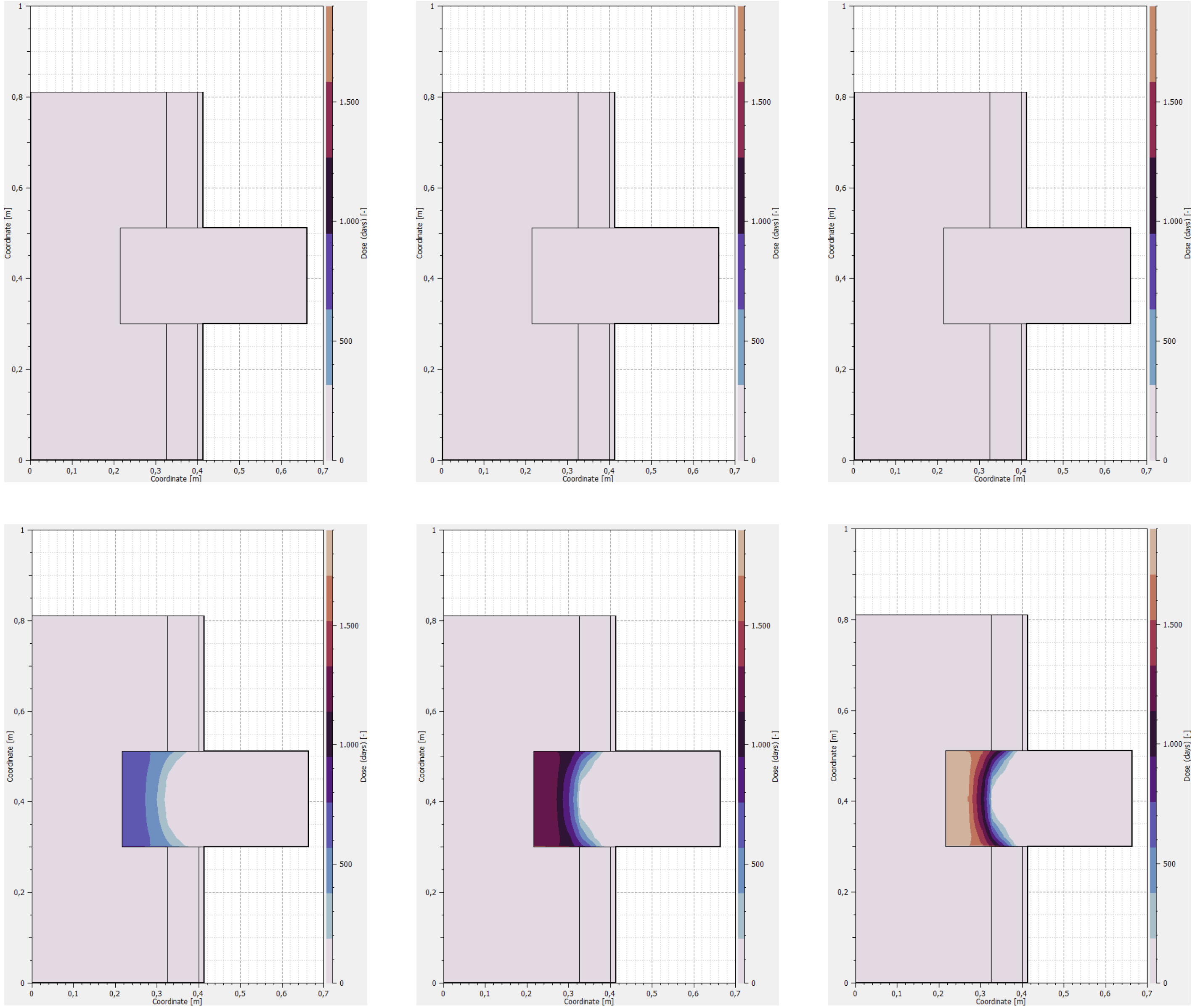
Dose after 4 years | Dose after 7 years | Dose after 10 years

Results

Orientation

East

West



Dose after 4 years

Dose after 7 years

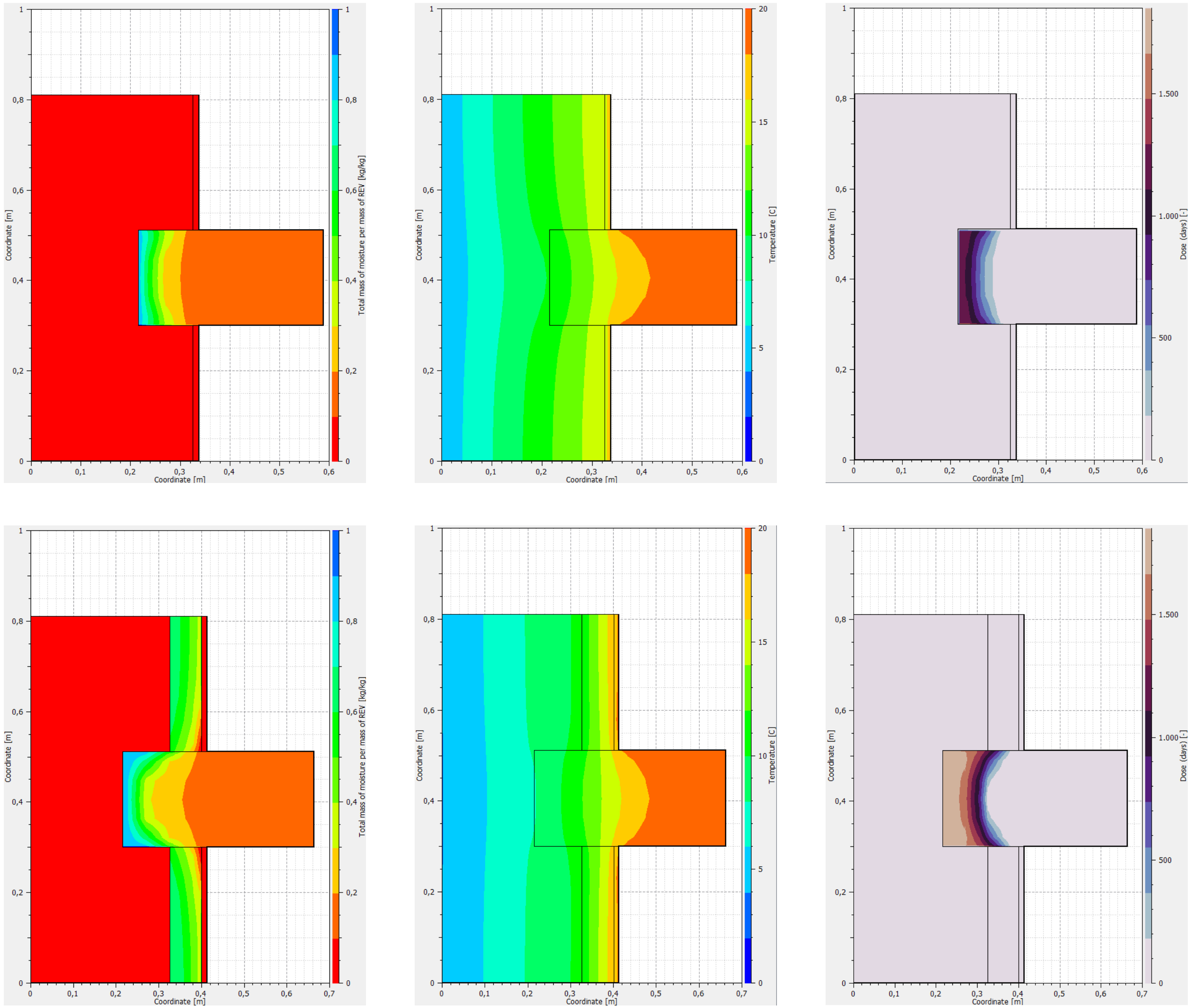
Dose after 10 years

Results

Pre / post insulation

Pre

Post



Moisture content

Temperature

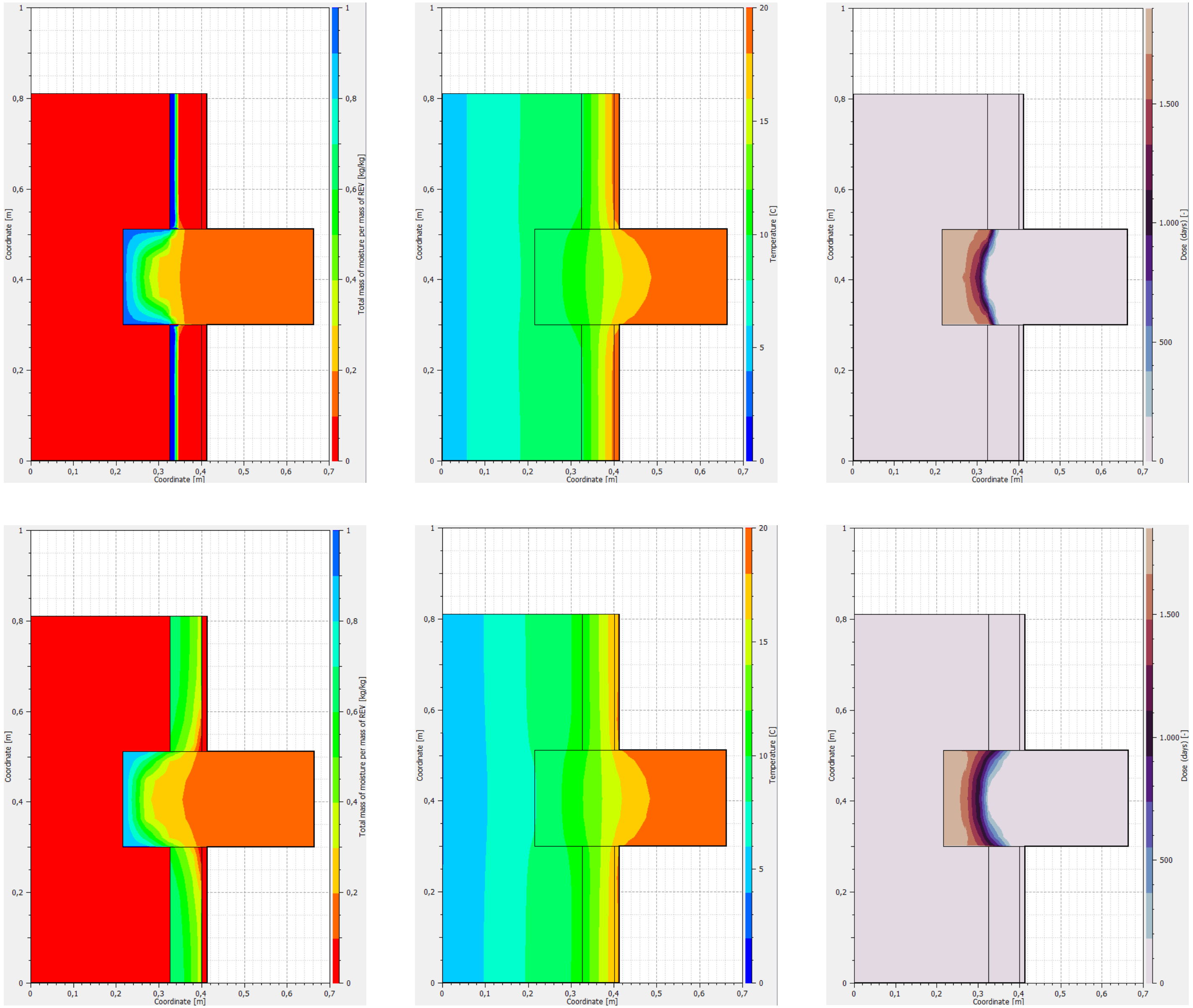
Dose

Results

Material type

Standard

Capillary active



Moisture content | Temperature | Dose

Conclusion

What influences the hygrothermal safety of embedded wooden beam in solid brick masonry with interior capillary active insulation most and how to mitigate the risk of wood decay?

Conclusion

- Exterior boundary conditions, such as rain load, solar radiation and orientation are more dominant on hygrothermal performance of the geometry

- Material properties are of secondary importance. Revealing positive contribution to the prevention of wood decay but incapable of reducing moisture content to levels that prevent wood decay completely

Recommendation

- Further research to gain more knowledge on the influence of the boundary conditions on wood decay
- Further research the effect of material properties and geometry changes on the hygrothermal performance of the embedded wooden beam
- Improve wood decay assessments by automatic template files and wood coordinates selection scripts and add species database

Master Thesis

Building Technology
Graduation Studio

P5 presentation
January 13, 2025

Rik Trip
5421195

TU Delft | abt adviseurs in bouwtechniek

Methodology – Bounds (additional pages)

- Bounds:
  - Thermal conductivity (*TC*)

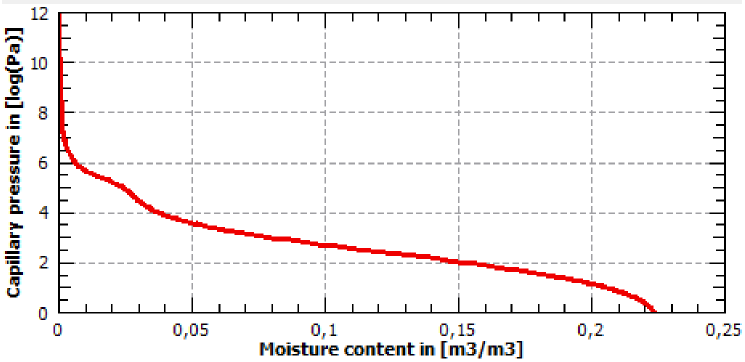
    Insulation *0.01875 – 0.075*
  - Water vapour diffusion resistance factor (*WVDRF*)

    Insulation *1 - 100*

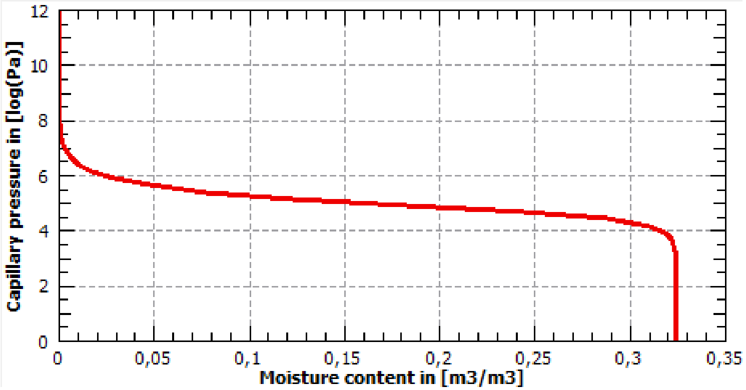
    Brick *5 – 50*
  - Pore size distribution factor (*NV*)

    Insulation *1.5 – 2.75*

    Brick *1.2 – 2*

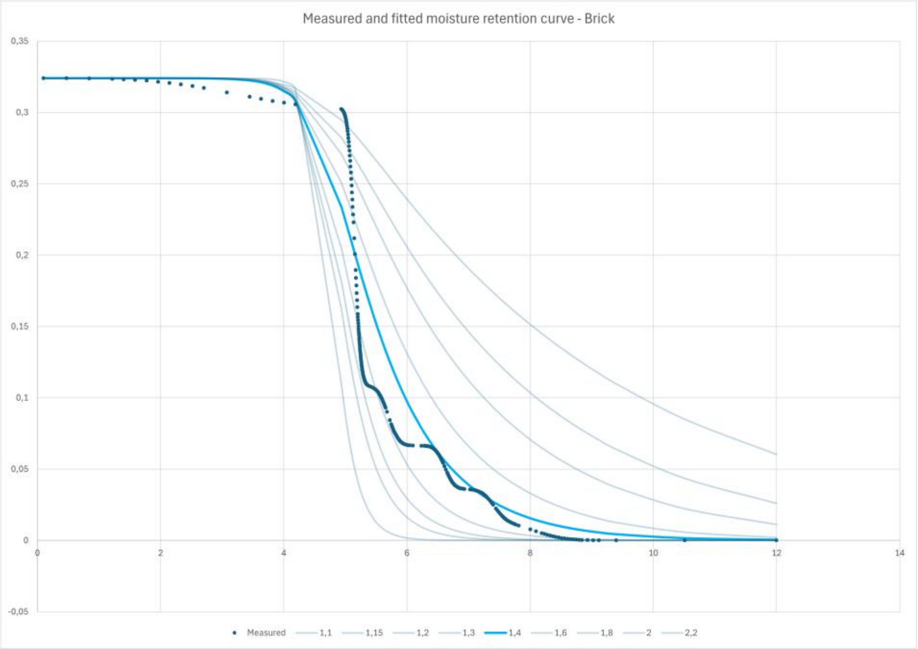


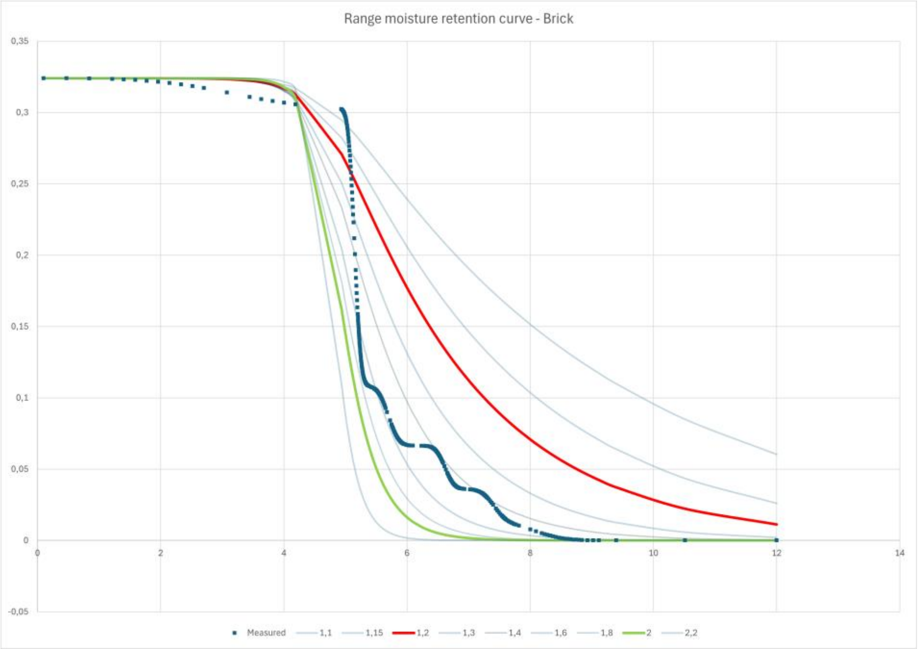
$$\theta(\psi) = \theta_r \frac{\theta_s - \theta_r}{[1 + (a|\psi|)^n]^{1-1/n}}$$



Methodology – Bounds (additional pages)

- Brick fitted curve and sensitivity analysis range





Methodology – Bounds (additional pages)

- Insulation fitted curve and sensitivity analysis range