

# Final Project Report

Andres Calvo (ac228)

Tue., Apr. 30

```
1 using Revise
2 using HouseElevation
3
4 using CSV
5 using DataFrames
6 using DataFramesMeta
7 using Distributions
8 using LaTeXStrings
9 using Metaheuristics
10 using Plots
11 using PlotThemes
12 using Random
13 using Unitful
14 using Extremes
15 using StatsPlots
16
17 Plots.default(; margin=5Plots.mm)
18 theme(:dao, grid = false,
19         minorgrid = false,
20         tick_direction = :out,)
```

## 1 Introduction

There is a great interest in modeling extreme weather events considering the large exposure of physical infrastructure, frequent occurrence of natural disaster and climate change. The sea level rise in particular is of the outmost interest for industrial communities located in hurricane prone regions such as Galveston and Houston Gori et al. (2020), Bernier and Padgett (2019), Amini and Padgett (2023). The present work leverages the class final project to consider sequential analysis of risk mitigation of industrial structures exposed to storm surges. Sequential actions (i.e. elevating) on structures can be performed by allowing state variables to evolve in time and a set of adaptive policies that trigger or not the mitigation decision. Closely based on the proposed work by Garner and Keller Garner and Keller (2018), two types of adaptive policies are considered and two profiles (respectively) are explored including the Do-nothing case. The two considered profiles are the proactive and reactive decision maker. The former is related to a decision profile where a trigger decision happens regardless of any perceived damage. In particular, the free-board height, the difference between the mean sea-level (MSL) and the inundation level of the building is used to

define the triggering event (e.g. if a defined free-board of 3 ft is reach by the MSL of year  $t$ , an elevation a policy is trigger). This freeboard is based on the work by Davis et.al Davis, Faber, and Stedinger (2008). In contrast, a reactive policy is based on a given threshold (i.e. maximum water level in year  $t$ ) that triggers a decision on elevating the building. The two profiles in addition to the Do-nothing case are analyzed. Different sources of uncertainty are considered including financial, physical and most importantly, the sea level rise.

## 1.1 Case study

The case study was chosen given the ongoing research on industrial structures in coastal communities in the US Gulf Coast. Therefore, a lightweight industrial building (i.e. warehouse) was selected in Galveston, Tx close to the port. The structure consists in a steel frame structure with a steel casing and roof. The exact use of the building is not known but some expensive machinery and contents are expected inside the building that are exposed to storm surges and to experiment damages. Figure 1 shows the building location and characteristics. The building is located 3100 ft apart from a NOAA methodological station and a sea level gauge in Pier 21 (station 8771450). The current elevation of the building respect to the gauge is close to 4 ft, nevertheless, this offset is going to be treated as a variable in order to compare different policies.

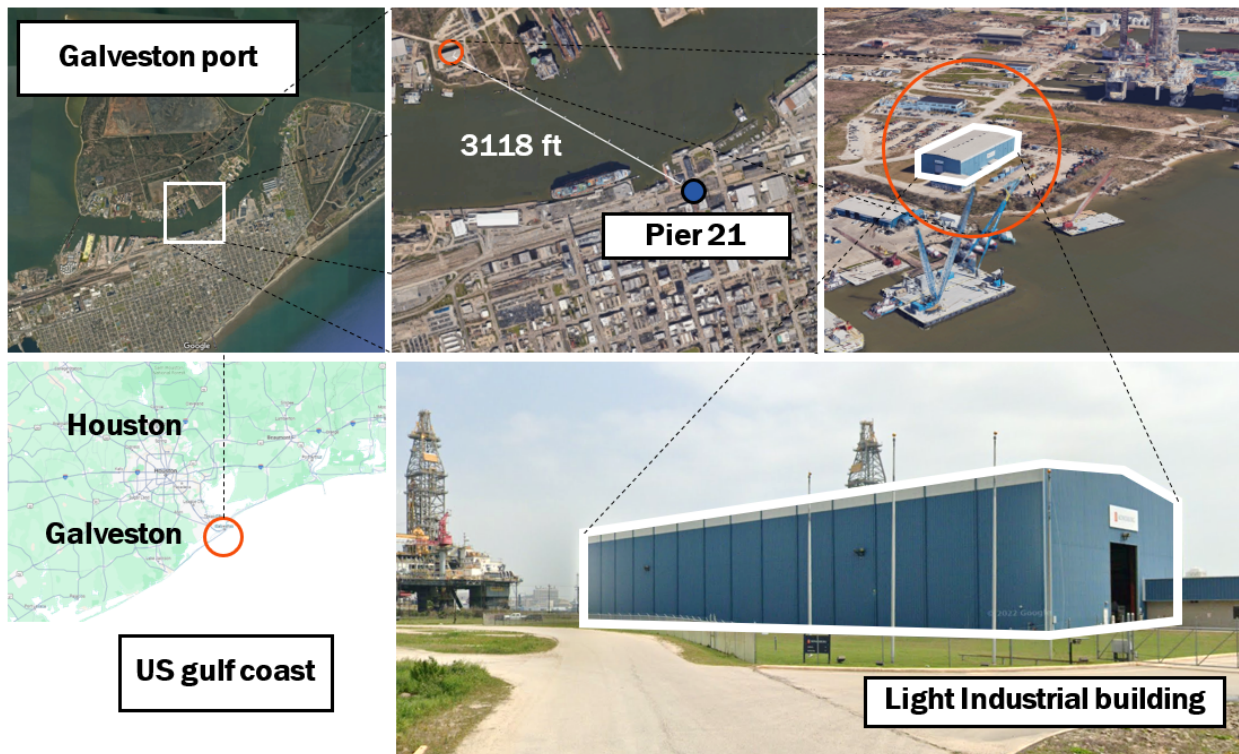


Figure 1: Figure 1

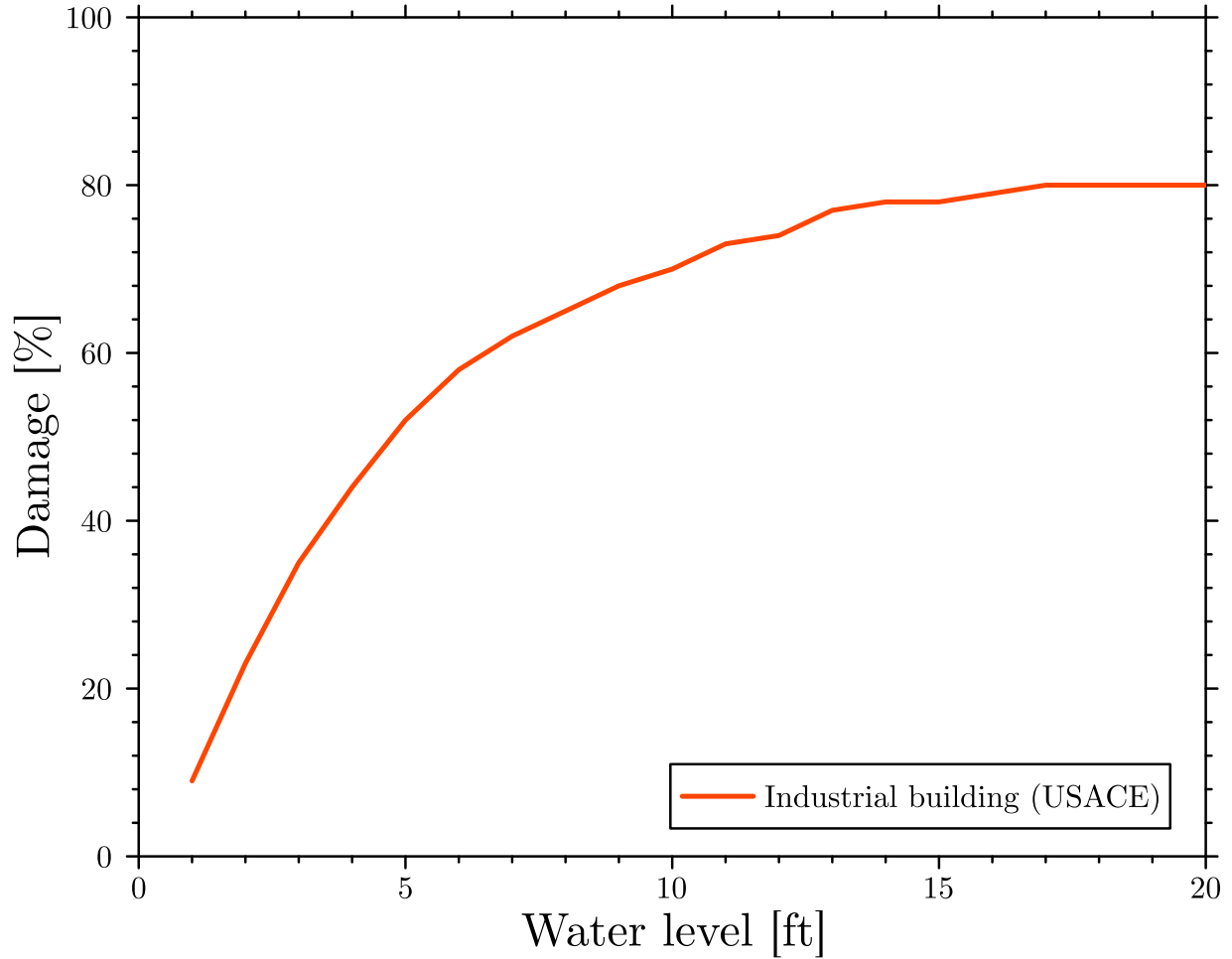
## 1.2 Depth-Damage curve

The damage of the structure is going to be assessed using a defined depth-damage curve (DDC) developed by Army Corps of Engineering (USACE) specifically for the Galveston context. In the HAZUS manual FEMA (2022), the curves are applicable to freshwater flooding, under slow-rise,

slow-recession and little velocity which is not the case of storm surges and hurricane surges in particular. Nevertheless, the USACE developed the curves given the Galveston exposure to tidal waves and hurricane storm surges, therefore, they are used in the present work. The “Average Light Industrial, Contents (Equipment/ Inventory)” DDC was selected from the available curves and is shown in Figure 2a. It was selected considering the equipment and contents that represent the largest portion of losses due to damages. The is large uncertainty in this model that is not manage or propagated. As explain by empirical big data Wing et al. (2020), the losses in residential houses follow non-monotonically increasing functions and rather unexpected distributions (e.i. distribution). Nevertheless, the DDC will be treated deterministically given that there is not available data to propagate the uncertainty. However, the DDC will not be a state variable and all simulations will be using the same DDC. This way, results will be consistent. The valuation of the building of 3’445.000 USD was done using commercial data from APX and a structure to content ratio of 1:8.

```

1 offset = 4
2 building = let
3     haz_fl_dept = CSV.read("data/haz_fl_dept.csv",DataFrame)
4     desc = "Average Light Industrial, Contents (Equipment/Inventory)"
5     row = @rsubset(haz_fl_dept, :Description == desc)[1,:]
6     area = 33_000u"ft^2"
7     height_above_gauge = (offset)u"ft"
8     House(
9         row;
10        area = area,
11        height_above_gauge = height_above_gauge,
12        value_usd = 4_000_000,
13    )
14 end
15 plot(building.ddf.itp[:,
16     xlabel = "Water level [ft]",
17     ylabel = "Damage [%]",
18     ylims = [0,100],
19     xlims = [0,20],
20     label = "Industrial building (USACE)",
21     legend = :bottomright,
22     size = (500, 400),
23     color = "orangered",
24     linewidth = 2,)
```



### 1.3 Sea-Level model

The model proposed by Oddo et al. Oddo et al. (2020) is used to generate stochastic realizations of the sea-level paths from 2024 to the time horizon T. This model will be used to represent the mean-sea-level (MSL) of a given year. Variations within the year are not considered. There is no probabilistic information to suggest the frequency of each scenario or the probability of occurrence. Therefore, random sampling within a Monte Carlo simulation is used to considered the average effect of the sea level rise. This is considered valid within the exploratory modeling context of the present work. Some realizations and statistics of these are represented in Figure 2b.

```

1 slr_scenarios = let
2   df = CSV.read("data/slr_oddo.csv", DataFrame)
3   [Oddo17SLR(a, b, c, tstar, cstar) for (a, b, c, tstar, cstar) in eachrow(df)]
4 end
5
6 case_scenarios = let
7   years = 1900:2150
8   s_level = zeros(size(slr_scenarios)[1],size(years)[1])
9   count = 1
10  for s in slr_scenarios

```

```

11         s_level[count,:] = s.(years)
12         count +=1
13     end
14     q_stat = [0.05, 0.25, 0.50, 0.75, 0.95]
15     scenarios = zeros(size(years)[1],size(q_stat)[1])
16     for q in 1:size(q_stat)[1]
17         scenarios[:,q] = [quantile(s_level[:,year],q_stat[q]) for year in 1:size(years)[1]]
18     end
19     scenarios
20 end

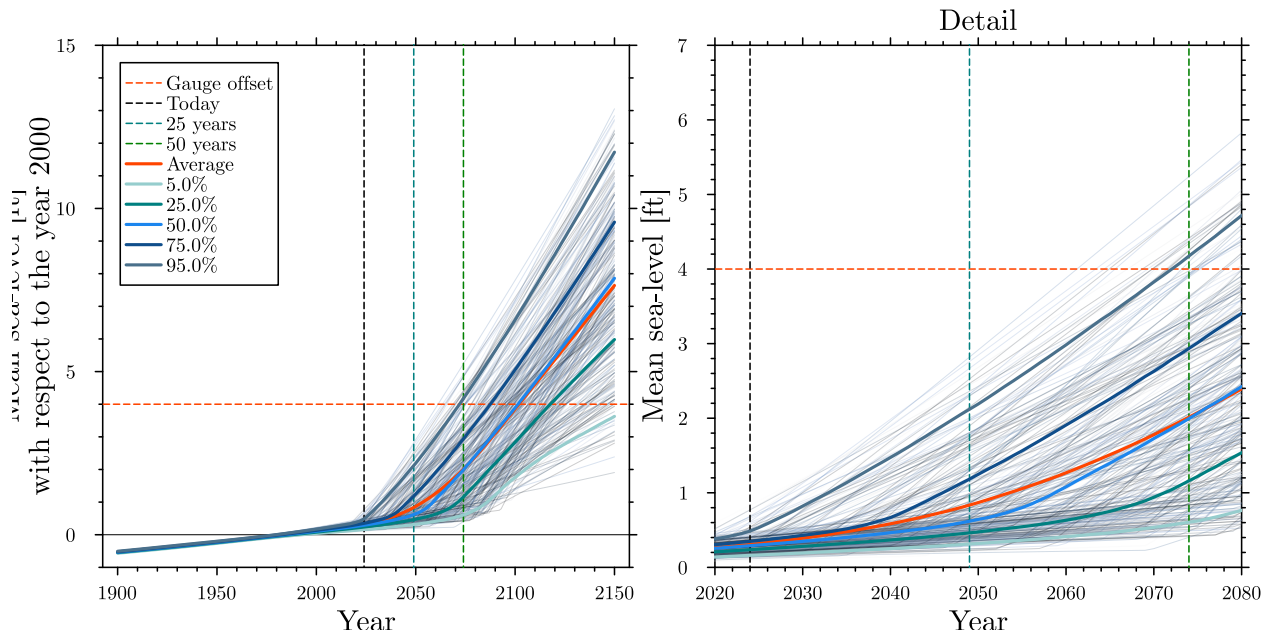
1 let
2     years = 1900:2150
3     p = plot(;
4         xlabel = "Year",
5         ylabel = "Mean sea-level [ft]\nwith respect to the year 2000",
6         label = "Oddo et al. (2017)",
7         legend = :topleft,
8         size=(400, 400),)
9     s_average = years.*0
10    N_samples = 300
11    for s in rand(slr_scenarios,N_samples)
12        plot!(p,
13            years,
14            s.(years);
15            palette = :oslo,
16            alpha = 0.2,
17            linewidth = 0.5,
18            label = nothing,
19            )
20        s_average += s.(years)
21    end
22    s_average /= 300
23
24    hline!([0]; color = "black", linewidth = 0.50, label = nothing)
25    hline!([offset]; color = "orangered", linewidth = 1, label = "Gauge offset", style = :dash)
26    vline!([2024]; color = "black", linewidth = 1, label = "Today", style = :dash)
27    vline!([2024+25]; color = "teal", linewidth = 1, label = "25 years", style = :dash)
28    vline!([2024+50]; color = "green", linewidth = 1, label = "50 years", style = :dash)
29    plot!(years,
30        s_average;
31        ylims = [-1,15],
32        color = "orangered",
33        label = "Average",
34        linewidth = 2,
35        )
36

```

```

37 colors_p = ["paleturquoise3","teal","dodgerblue2","dodgerblue4","skyblue4"]
38 q_stat = [0.05, 0.25, 0.50, 0.75, 0.95]
39 for q in 1:size(q_stat)[1]
40 plot!(p,years,case_scenarios[:,q];
41       label = "$(q_stat[q] * 100)%",
42       linewidth = 2,
43       color = colors_p[q])
44 end
45 pl_det = plot(p;
46              ylims = [0,7],
47              ylabel = "Mean sea-level [ft]",
48              xlims = [2020,2080],
49              title = "Detail",
50              legend = false)
51
52 plot(p, pl_det, layout = 2,size = (800, 400))
53 end

```



## 1.4 Storm surge hazard

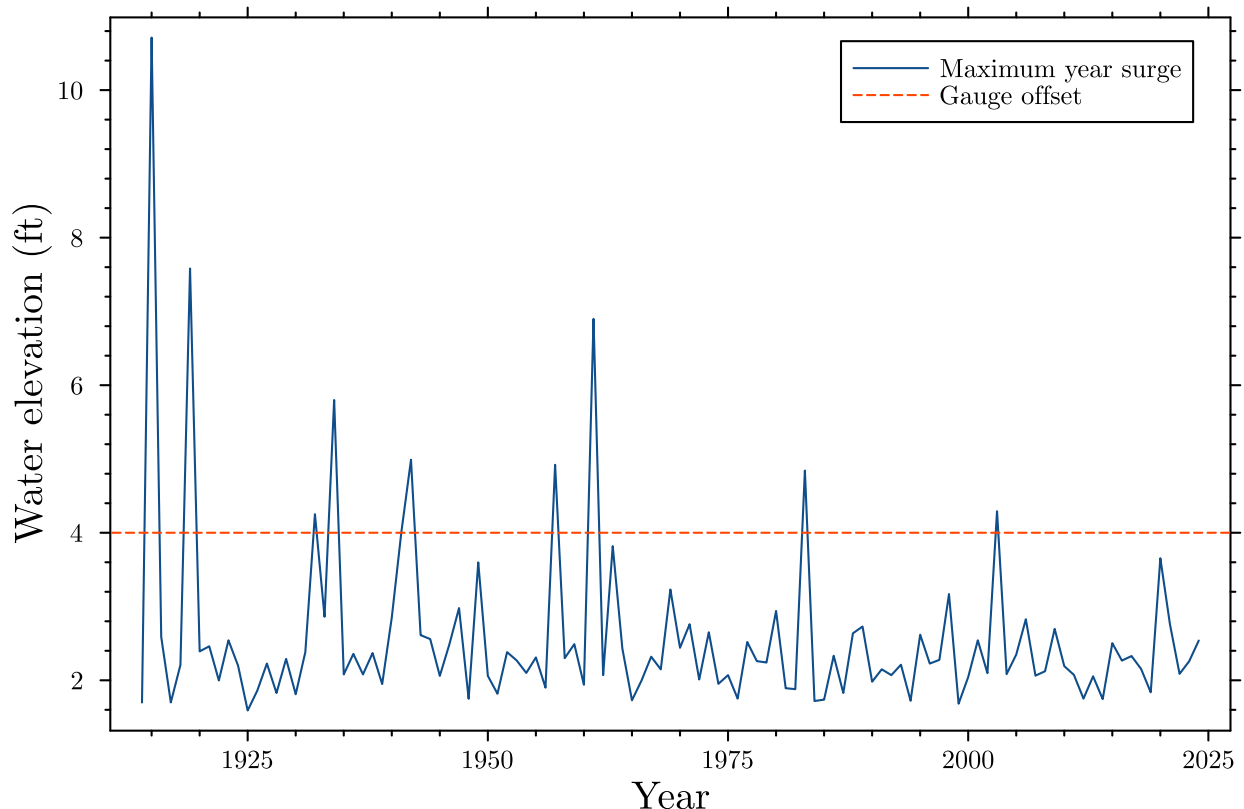
The hazard model for storm surges is represented by an Extreme Value Distribution (EVD) with parameters fitted to the available data of the gauge. Similar work has been done using NOAA data, for example, Buchanan et al. recommend using a Generalized Pareto Distribution with in the extreme value analysis Buchanan et al. (2016). There are different approaches that could be used as hazard model including stochastic simulation, existing models (e.g. ADCIRC+SWAN) or a worst-case scenario. Nevertheless, the uncertainty in the surges above the MSL are sampled from the EVD randomly. To that end, maximum and mean sea levels for the Pier 21 gauge are used. The surge data per year to be assigned to the state variable correspond to the year-maximum level over the MSL. The available data contains 1288 records from 1914 to the date. The MSL is subtracted

from the highest datum in ft, and the year maximum to represent the year surge. The package Extremes.jl and the function gevfit are used to fit the GEV parameters. For the year maximum surge, the parameters are found to be  $\mu=2.59$ ,  $\sigma=0.51$  and  $\xi=0.38$ . Figure 3 show the maximum surge per year and the fitted GEV distribution and shows the record from 1914 where is clear the linear trend of sea level rise in Galveston. The largest peaks in the record correspond to past hurricane events including the 1915 Galveston hurricane and 1963 Cindy.

```

1 water_max = CSV.read("8771450_max_year.csv",DataFrame)
2 let
3     p = plot(water_max.Year,
4               water_max.MaxtoMLS;
5               xlabel = "Year",
6               ylabel = "Water elevation (ft)",
7               label = "Maximum year surge",
8               legend = :topright,
9               color = "dodgerblue4",
10              linewidth = 1,)
11     hline!([offset]; color = "orangered", linewidth = 1, label = "Gauge offset", style = :dash)
12     p
13 end

```



```

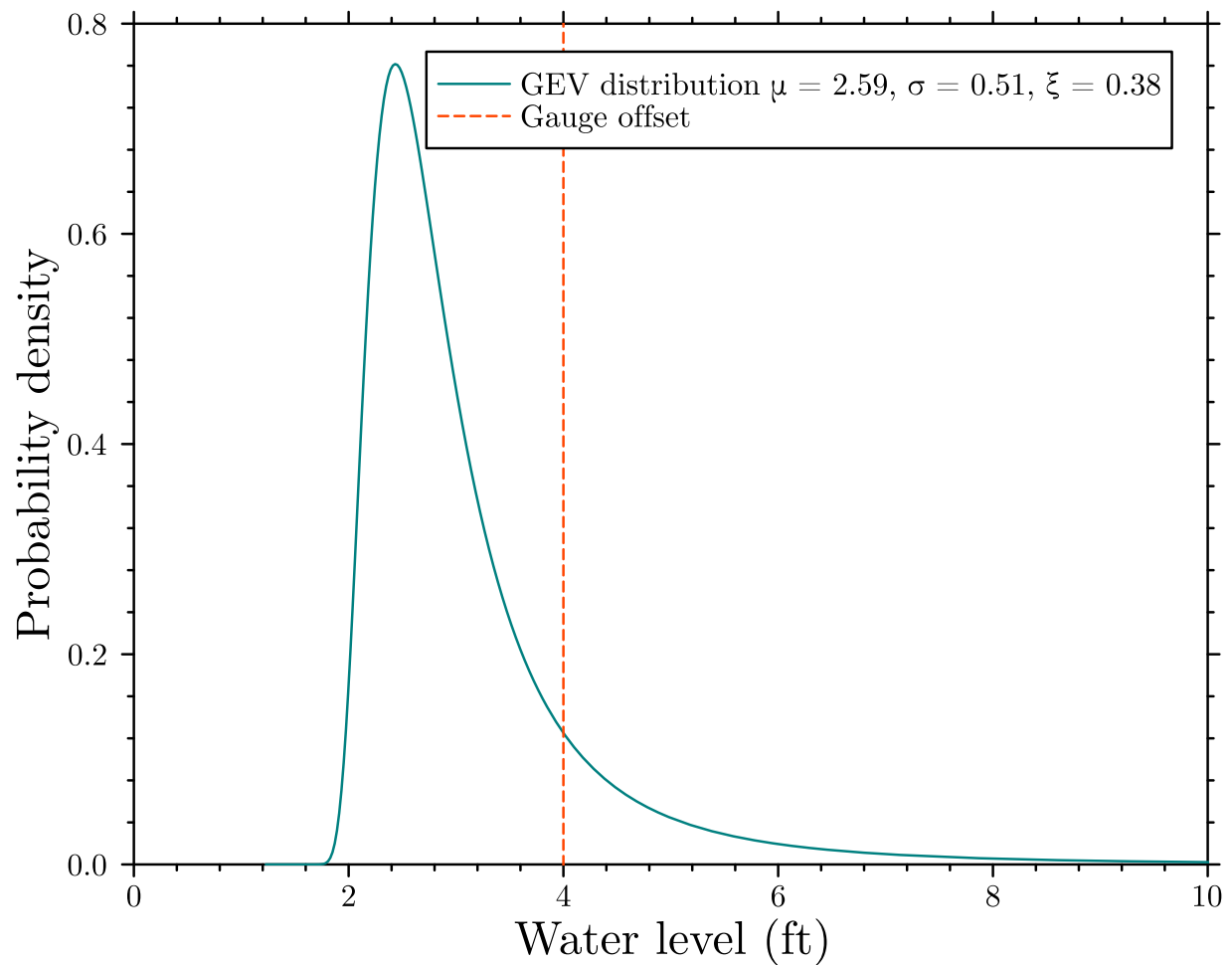
1 = gevfit(water_max, :LevelMaxYear).^
2 let
3     = [1]
4     = exp([2])

```

```

5     = [3]
6
7     p = plot()
8     plot!(p, GeneralizedExtremeValue( , , );
9         xlabel = "Water level (ft)",
10        ylabel = "Probability density",
11        ylims = [0,0.80],
12        xlims = [0,10],
13        label = "GEV distribution  $\mu = \$(round( , digits = 2))$ ,  $\sigma = \$(round( , digits = 2))$ ,  $\xi = \$(round( , digits = 2))$ ",
14        legend = :topright,
15        size = (500, 400),
16        color = "teal",
17        linewidth = 1,)
18     vline!([offset]; color = "orangered", linewidth = 1, label = "Gauge offset", style = :dash)
19 end

```



```

1 function draw_surge_distribution()
2     = [1]
3     = exp( [2])

```



```

4     = [3]
5     GeneralizedExtremeValue( , , )
6 end

```

draw\_surge\_distribution (generic function with 1 method)

## 1.5 Discount rate

The discount rate (DR) has been found to be very impactful to the discounted cashflows in the future. There is no information or time histories of financial data to be able to model a future behavior. There are not enough arguments to choose a number over other for industrial warehouses in the Galveston port. Nevertheless, in order to capture the uncertain behavior two discount rate scenarios are considered for the state of the world (SOW) of the simulations. The first is a low DR of 3.0% (low respect to the cost of opportunity of an industrial facility) and high DR of 7.0%, relative to the 3%. The exploratory nature of the analysis allows to simulate the effect of the DR and support decision making.

```

1 function draw_discount_rate(level)
2     if level == "low"
3         d_rate = 0.03
4     elseif level == "high"
5         d_rate = 0.07
6     end
7     return d_rate
8 end
9 draw_discount_rate("high")

```

0.07

## 1.6 Time horizons

The time frame is also uncertain and very impactful in terms of cost-benefit analysis. Most buildings nowadays are design to life spans of 50 years to 100 years. Nevertheless, some project horizons (and in particular for an industrial facility) can be much lower (e.g. 10 to 20 years). Therefore, two SOW's are proposed to explore the effect of the time frame. A short 25-year and a long 50-year horizons are proposed. The consequences are very sensitive in terms of the SLR model that is non-linear in time. As seen in Figure 2a, the considerable number of SLR scenarios have an MSL above a 4-ft offset, that is, regardless of the surge, the building would experience flooding.

```

1 timeframe_25 = ModelParams(; house = building,
2                             years = 2024:2024+25)
3 timeframe_50 = ModelParams(; house = building,
4                             years = 2024:2024+50)

```

## 2 Adaptive policies

Two decision profiles (three consider the Do-nothing profile) are considered in the present analysis. A cost-benefit analysis (CBA) will be used to compare the results of each profile  $i$  in terms of the net present value (NPV) of the utility function  $u()$ .

$$NPV^i = \frac{1}{N} \sum_n \sum_{t=0}^T u(s_n, a_t^i, \mathbf{x}_t)(1 - \gamma_n)^t$$

Where  $N$  are the number of Monte Carlo simulations that allow for capturing an average behavior,  $T$  is the time horizon,  $s$  is the SOW (e.g. surge sampling) of the  $n$  simulation,  $a_t^i$  is the  $i$ -th policy profile elevation action in the time  $t$ ,  $\mathbf{x}_t$  is the state variable that updates every year (e.g. max year level, building elevation) and  $\gamma_n$  is the discount rate of the  $n$ -SOW simulation. Given that the policies trigger the action according to the state variable, two variables define the simulation: a triggering event and an elevation quantity.

For the proactive profile, the triggering event is associated with the free-board distance  $\Delta$ , that is, the level difference between the MSL and the inundation level of the building that is calculated for every time step  $t$ . It is considered proactive given that it is independent of a surge event. The reactive profile, in contrast, have a triggering event dependent of a surge level and a given maximum water level threshold  $L_{\max}$  that represents a flood event in itself. The amount of elevation is the second variable per profile. Some fixed values are shown in Table 1 to illustrate the different profiles according to a qualitative risk aversion metric.

Qualitative Risk aversion	Triggering event		Elevation action [ft]
	<i>Proactive</i> $\Delta$ [ft]	<i>Reactive</i> $L_{\max}$ [ft]	
Do-nothing	$-\infty$	$+\infty$	0
Low	1	8	[0-14]
Mid-Low	2	6	[0-14]
Mid-High	3	4	[0-14]
High	4	2	[0-14]

Figure 2: Table 1

```

1 mp = timeframe_50
2 sows = let
3     N_SOW = 10_000
4     sows = [SOW(rand(slr_scenarios),
5                 draw_surge_distribution(),
6                 draw_discount_rate("low")) for _ in 1:N_SOW]
7 end
8 function objective_function_Pro(a::Vector{Float64})
9     elev = a[1]
10    elev = Action((elev)u"ft")
11    delta = a[2]
12    NPV = [simulate(s,elev,mp,ProactivePolicy(delta)) for s in sows]./1e6
13    return mean(NPV)
14 end
15 function objective_function_Re(a::Vector{Float64})
16    elev = a[1]
17    elev = Action((elev)u"ft")
18    level = a[2]
19    NPV = [simulate(s,elev,mp,ReactivePolicy(level)) for s in sows]./1e6

```

```

20     return mean(NPV)
21 end
22
23 [objective_function_Pro([8.0,3.0]) , objective_function_Re([8.0,2.0])]

```

```

2-element Vector{Float64}:
 3.5408812559727814
 3.6305353642493863

```

## 2.1 Code implementation

The class code has been modified for allowing sequential decision making, state variables, time-step simulation, and simulation. The overall code structure was based on the sequential lab for the Garage building. The building state variable was defined in core.jl as follows:

```

1  # BuildingState x is a state variable that saves each time step information
2  mutable struct BuildingState{T<:AbstractFloat}
3      elevation::T
4      year::Int
5      MSL::T
6      Damage::T
7      max_level::T
8  end
9
10 # The function is initialized in zeros except for the year in 1
11 function BuildingState()
12     return BuildingState(0.0, 1, 0.0, 0.0, 0.0)
13 end

```

The building action and adaptive policies were also defined in core.jl

```

1  # The building action is an elevation increment Δ
2  struct BuildingAction
3      Δelevation::AbstractFloat
4  end
5
6  # Two types of abstract policies are defined
7  abstract type AbstractPolicy end
8  # The proactive policy uses a Free-board distance to trigger a decision sequence
9  struct ProactivePolicy <: AbstractPolicy
10     delta_min::AbstractFloat
11 end
12 # The reactive policy uses a year maximum water level to trigger a decision sequence
13 struct ReactivePolicy <: AbstractPolicy
14     critical_level::AbstractFloat
15 end

```

The policy functions were defined in run\_sim.jl

```

1 # The proactive function triggers an elevation action only if the freboard distance  $\Delta$  is surpa
2 function get_ProactiveAction(x::BuildingState, policy::ProactivePolicy, a::Action)
3     if (x.elevation - x.MSL) < policy.delta_min
4         return BuildingAction(a. $\Delta$ _ft)
5     else
6         return BuildingAction(0)
7     end
8
9 end
10
11 # The proactive function triggers an elevation action only if the maximum level threshold Lmax
12 function get_ReactiveAction(x::BuildingState, policy::ReactivePolicy, a::Action)
13     if x.max_level > policy.critical_level
14         return BuildingAction(a. $\Delta$ _ft)
15     else
16         return BuildingAction(0)
17     end
18
19 end

```

The time step function was defined to estimates annual losses based on the maxim year surge

```

1 # One time step in the sequential decision siluation
2 function run_timestep(
3     x::BuildingState, sow::SOW, a::Action, p::ModelParams, policy::T
4 ) where {T<:AbstractPolicy}
5
6     # Initial level of the building
7     if x.year == 1
8         x.elevation = p.house.height_above_gauge_ft
9     end
10
11     # Mean sea level from the SLR model
12     x.MSL = sow.slr(p.years[x.year])
13
14     # Set the elevation action according to a policy
15     if policy isa ProactivePolicy
16          $\Delta$  = get_ProactiveAction(x, policy, a)
17     elseif policy isa ReactivePolicy
18          $\Delta$  = get_ReactiveAction(x, policy, a)
19     end
20
21     # Estimates the construction cost of the elevation action
22     construction_cost = elevation_cost(p.house,  $\Delta$ . $\Delta$ elevation)
23
24     # Change the elevation state of the building
25     x.elevation +=  $\Delta$ . $\Delta$ elevation
26

```

```

27     # Calculate the maximum yearly level or year flood event
28     x.max_level = rand(sow.surge_dist) + x.MSL - x.elevation - Δ.Δelevation
29     # Estimates the building damages and losses
30     damages_year = p.house.ddf(x.max_level)/100
31     x.Damage = damages_year
32     EAD = x.Damage * p.house.value_usd
33     # The year cost cashflow is the sum of the construction losses costs
34     cost = EAD + construction_cost
35
36     return cost
37 end

```

The simulation of the T years is defined as follows:

```

1  function simulate(sow::SOW, a::Action, p::ModelParams, policy::T) where {T<:AbstractPolicy}
2
3     # initialize the model
4     x = BuildingState()
5
6     # Calculate the year cashflow
7     years = collect(1:size(p.years)[1])
8     costs = map(years) do year
9         x.year = year
10        run_timestep(x, sow, a, p, policy)
11    end
12
13    # Calculate the net present value of the simulation
14    discount_weights = @. (1 - sow.discount_rate)^(years - 1)
15    npv = sum(costs .* discount_weights)
16    npv_millions = npv
17
18    return npv_millions
19 end

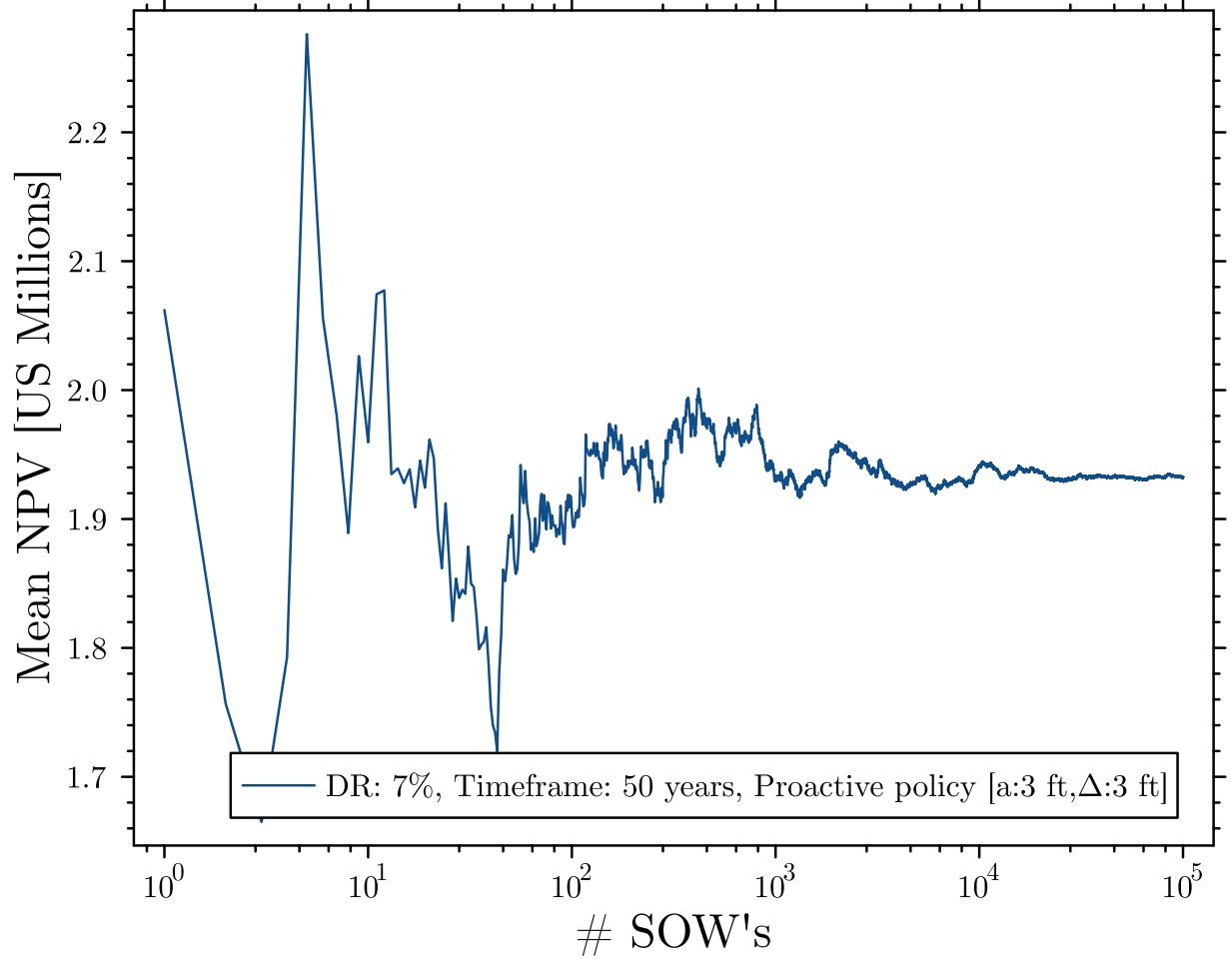
```

## 2.2 Monte Carlo simulations analysis

The first parameter to calibrate is the number of simulations to performed. Given that each simulation is sampling a maximum year surge, each simulation yields very different NPV. This differ with the case in which Monte Carlo simulations were performed within each time step. In order to capture the maximum year surge, this averaging process is no possible. One limitation, therefore, is assumed only one flood event per year. The Figure 5 shows how the average of the mean NPV is converging with the increasing number of simulations. This number determine whether or not an optimization routine if computationally cost effective. After 1,000 simulations the model varies the results in “±” 50,000 US dollars and after 10,000 simulations the difference is almost negligible. For 100,000 simulation there is complete visual convergence but this number of simulations would be prohibited for optimization purposes. The simulations represent a proactive policy with a free-board distance of 3 ft and an elevation action of 3 ft, DR of 7.0% and 50-year horizon. For example, using 10,000 simulation and a sample space of different triggering events,

elevation actions, and time frames, the same convergence behavior is observed.

```
1 let
2   N_samples = 100_000
3   sim = map(1:N_samples) do year
4     s = SOW(rand(slr_scenarios),
5              draw_surge_distribution(),
6              draw_discount_rate("high"))
7     a = Action((3)u"ft")
8     mp = timeframe_50
9     policy = ProactivePolicy(3)
10    simulate(s,a,mp,policy)
11  end
12  NPV_avg = zeros(N_samples)
13  for n in 1:N_samples
14    NPV_avg[n] = sum(sim[1:n])/n
15  end
16  plot(NPV_avg./1e6;
17       xlabel = "# SOW's",
18       xticks = ([1,10,100,1000,10000,100000]),
19       xaxis = :log,
20       ylabel = "Mean NPV [US Millions]",
21       label = "DR: 7%, Timeframe: 50 years, Proactive policy [a:3 ft,Δ:3 ft]",
22       legend = :bottomright,
23       size = (500, 400),
24       color = "dodgerblue4",
25       linewidth = 1,)
26 end
```



### 3 Sequential decision analysis

The state variable behavior is analyzed by sampling some simulations and plotting the year variables. Figure 6a shows 10 simulations for the proactive policy for 50-year horizon, high DR, 3 ft free-board and 4 ft action policy and 4 ft of offset. This means that if there is a year with a MSL of 1 ft ( $4\text{ftoffset} - 3\text{ft}\Delta$ ) a 4 ft elevation takes place. For high sea-level scenarios, the action happens rather quick (less than 10 years) but most of elevation actions happens in the 25 to 40 year. The simulations show damages for large surges given that the free-board is measure for the MSL not the maximum level that is unknown apriori. For extreme SLR scenarios, this profile elevates the building twice (one in the 6th year and again in the 48th year), however most elevation actions take place once.

In contrast, Figure 6b presents the reactive policy for same T and DR, but for a 3 ft elevation action and a Lmax of 2-ft. This means that after a year with a flood depth of 2 or more ft, the policy triggers an elevation of 3 ft. More action events are expected since they depend on the surges and not the seal level.

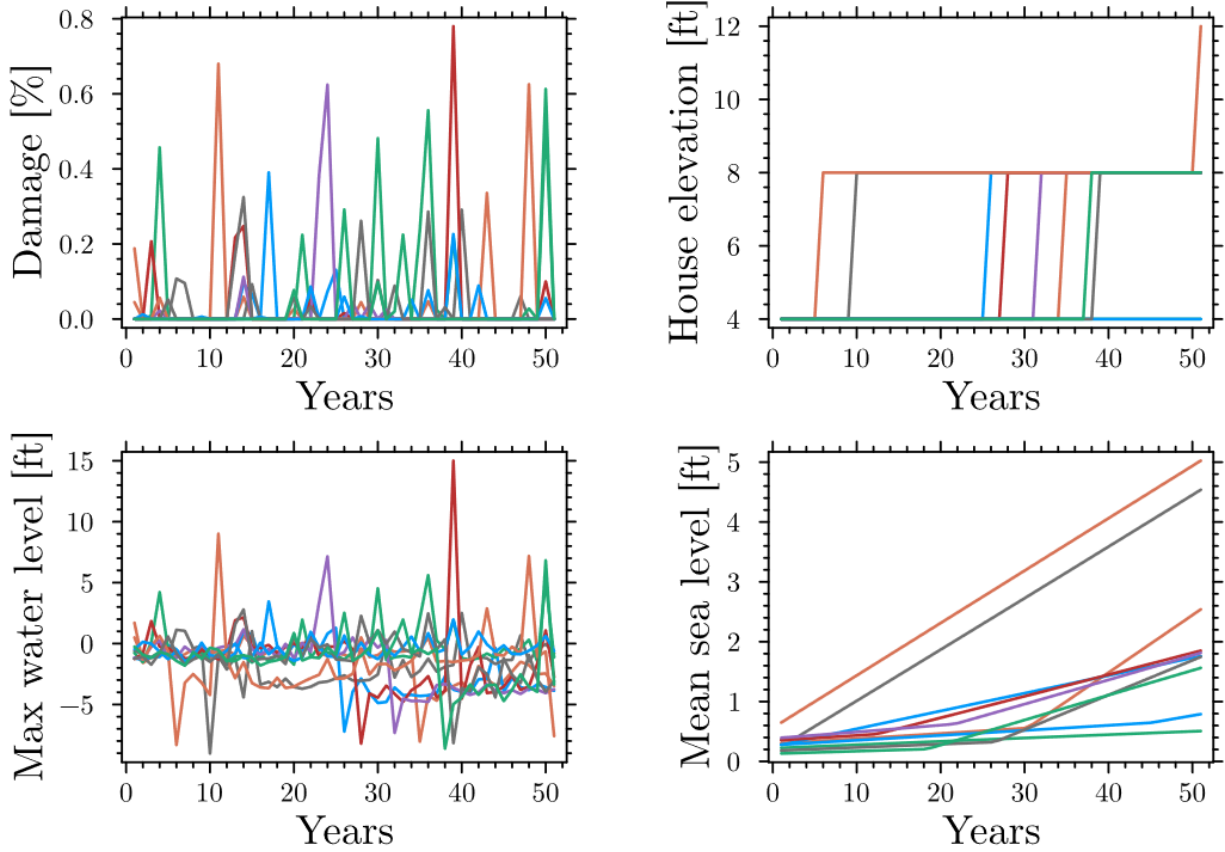


Figure 3: Figure 6a



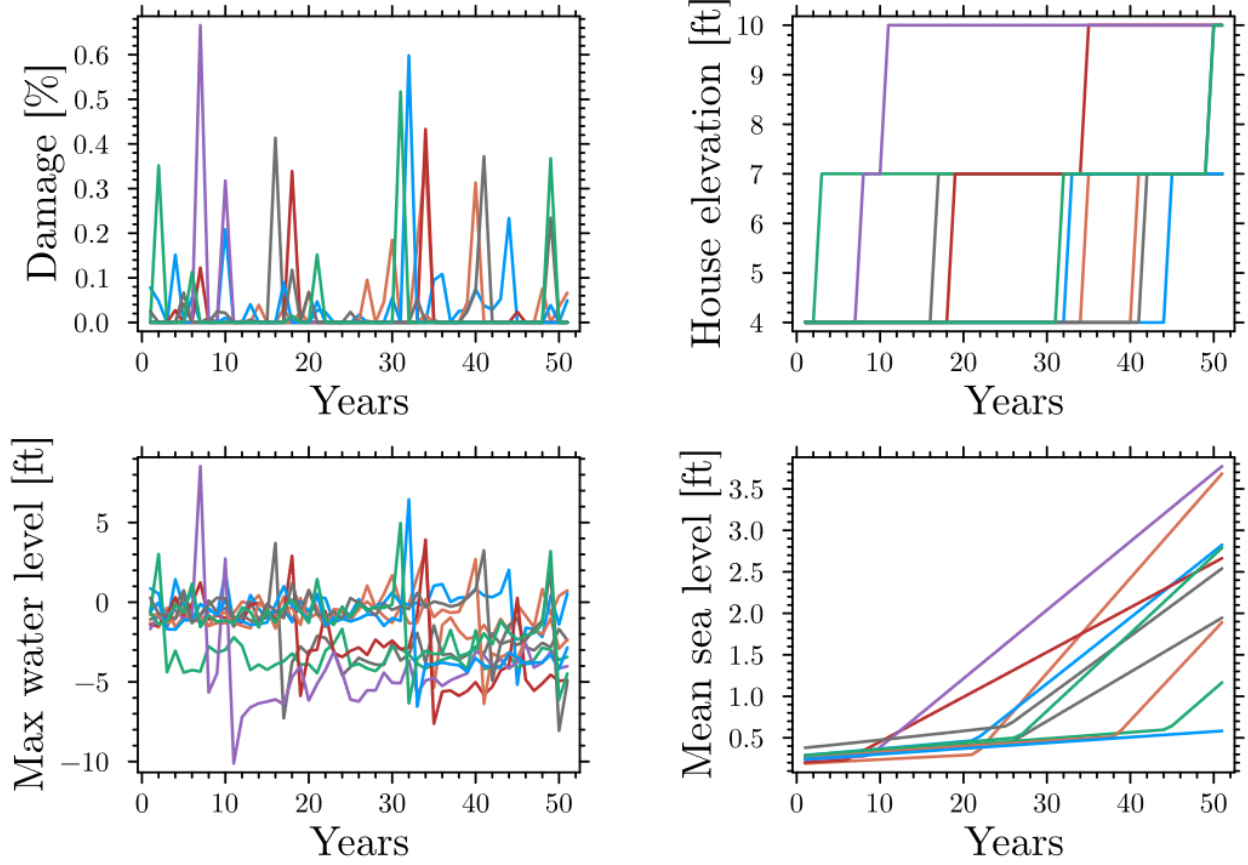


Figure 4: Figure 6b

### 3.1 Brute force optimization

The problem is set as a bivariate optimization, that is, the two variables are optimize, the triggering event (free-board or maximum water level) and the amount of elevation. The first approach is a Brute Force exploration optimization for a set of fixed decision policies. For the proactive and reactive policies, a discrete set of  $[1 - 10 \text{ ft}]$  in single feet increment (10 alternatives) was used for  $\Delta$  and  $L_{\max}$ , respectively. The sample space for elevation actions was 0 to 10 ft in a 2 ft increment (6 alternatives). The number of simulations was set in 10,000. Figure 7 shows how the average NPV changes for the different decision profiles, T and DR. Large DR decrease the present value of future damages (Figure 7a,b) therefore elevation actions are less cost-effective, this is particularly true for large time horizons that can be heavily impacted by the SLR. This is evident for low DR, where risk averse triggered events are cost-effective for a large time window but led to higher than Do-nothing NPV for short time frames (Figure 7c,d).

The initial offset of the building is a key parameter as well to make decisions of elevation. The brute force simulation was performed for buildings with 2 ft and 6 ft offset respect to the gauge. Figure 8 show the mean NPV for the least and most action prone scenarios, i.e. 25 year/7% and 50 year/3%, respectively, for the 2 ft and 6 ft offset buildings. For the former, having 2 to 4 ft of free-board led to optimal decisions for 6-8 ft elevation action range regardless of the T/DR. In contrast, for a 6 ft offset building, the do-nothing decision profile is virtually optimal for all scenarios.

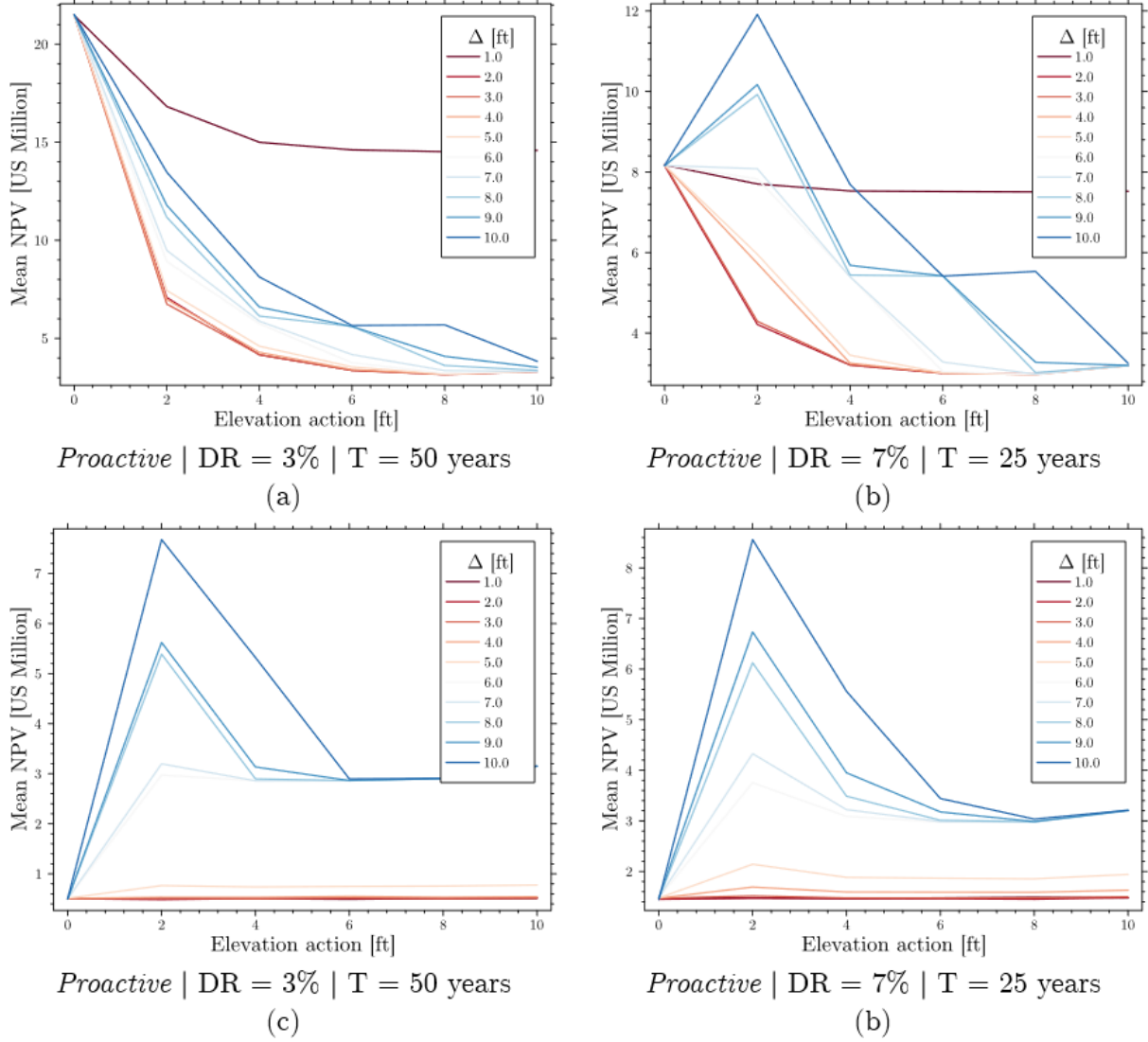


Figure 5: Figure 8

Regarding the decision profile policies, the proactive attitude can lead to higher NPV for risk averse triggering events (i.e. large free-board), however, for small free-board distances and for action prone scenarios (i.e. long-term analysis and low discount rate), the proactive policy yield to optimal decision making and minimize the NPV. The reactive policy in general lead to results with lower variability. This policy behavior is illustrated in Figure 9.

```

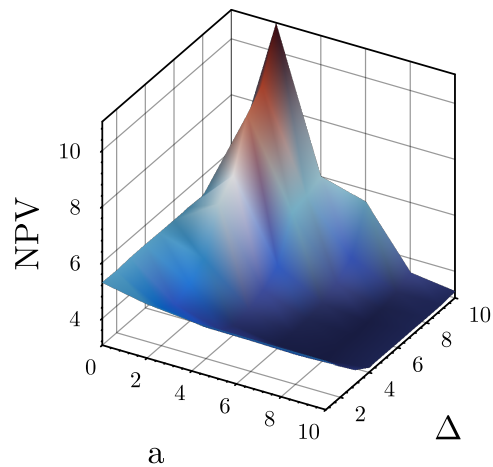
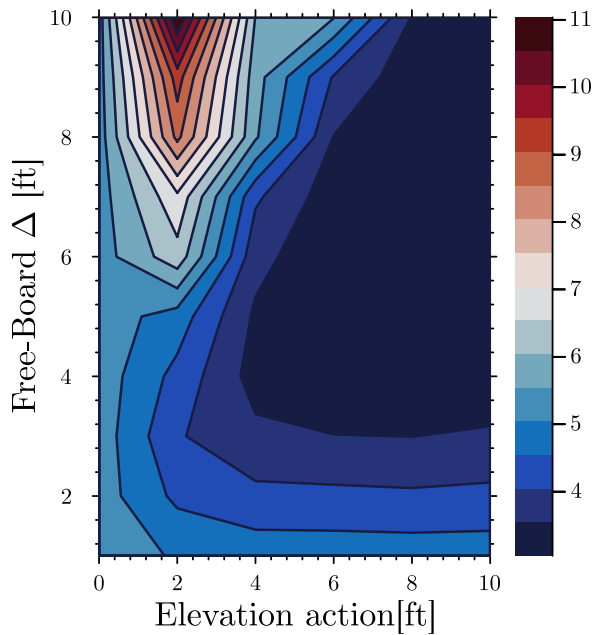
1 x1_pro = [0, 2, 4, 6, 8, 10.0]
2 x2_pro = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10.0]
3 z_pro = [objective_function_Pro([a,d]) for a in x1_pro, d in x2_pro]
4 let
5     P1 = plot(x1_pro,x2_pro,transpose(z_pro);
6             st = :contourf,
7             xlabel = "Elevation action[ft]",
8             ylabel = "Free-Board Δ [ft]",

```

```

9         ylabel = "Mean NPV [USD Millions]",
10        label = "Proactive policy",
11        legend = :outerbottom,
12        color = :balance,)
13
14    P2 = plot(x1_pro,x2_pro,transpose(z_pro);
15            st = :surface,
16            xlabel = "a",
17            ylabel = " $\Delta$ ",
18            ylabel = "NPV",
19            color = :balance,
20            cbar = false,
21            grid = true,
22            )
23    plot(P1,P2, layout = (1,2), size = (700,400))
24 end

```



```

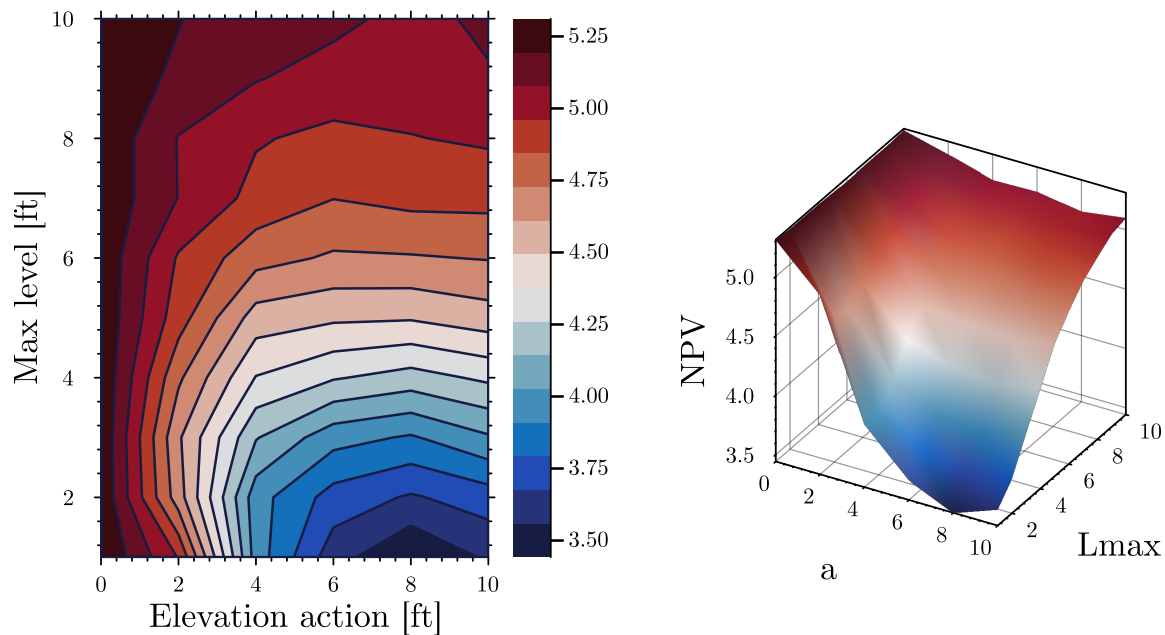
1  x1_re = [0, 2, 4, 6, 8, 10.0]
2  x2_re = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10.0]
3  z_re = [objective_function_Re([a,e]) for a in x1_re, e in x2_re]
4  let
5      P1 = plot(x1_re,x2_re,transpose(z_re);
6              st = :contourf,
7              xlabel = "Elevation action [ft]",
8              ylabel = "Max level [ft]",
9              ylabel = "Mean NPV [USD Millions]",
10             label = "Reactive policy",

```

```

11     legend = :outerbottom,
12     color = :balance,)
13
14     P2 = plot(x1_re,x2_re,transpose(z_re);
15             st = :surface,
16             xlabel = "a",
17             ylabel = "Lmax",
18             zlabel = "NPV",
19             color = :balance,
20             cbar = false,
21             grid = true)
22     plot(P1,P2, layout = (1,2), size = (700,400))
23 end

```



```

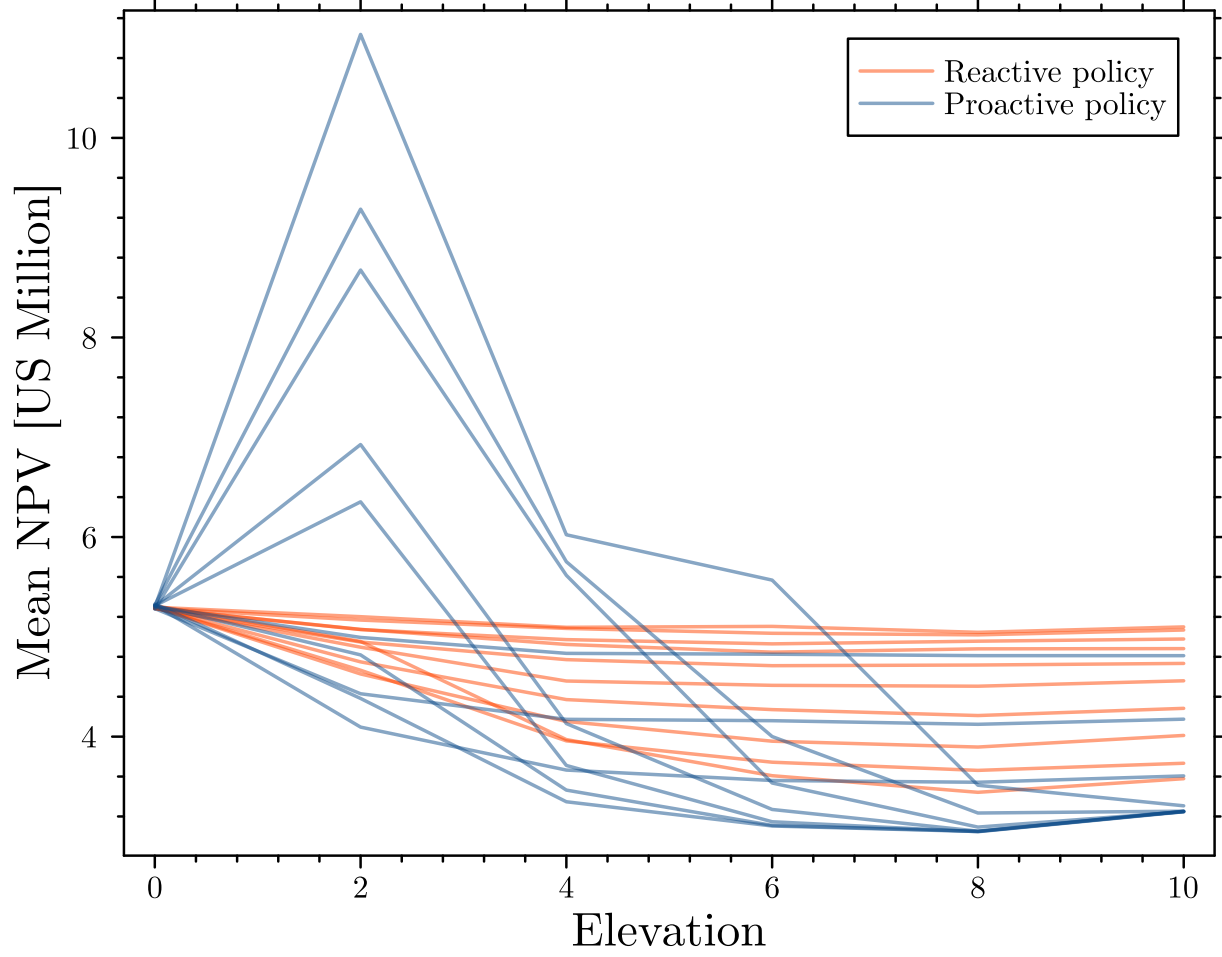
1 let
2     pl = plot(;xlabel = "Elevation action[ft]",
3             xticks = (0:2:10),
4             ylabel = "Mean NPV [US Million]",
5             size = (500, 400),
6             linewidth = 1,
7             legend = :topright,)
8     for level in 1:size(x2_re)[1]
9         if level == 1
10             plot!(pl,x1_re,transpose(z_re)[level,:];
11                 label = "Reactive policy",
12                 xlabel = "Elevation",
13                 color = "orangered",

```

```

14         alpha = 0.50,
15     )
16     else
17     plot!(p1,x1_re,transpose(z_re)[level,:];
18         label = nothing,
19         xlabel = "Elevation",
20         color = "orangered",
21         alpha = 0.50,
22     )
23     end
24 end
25 for delta in 1:size(x2_pro)[1]
26     if delta == 1
27     plot!(p1,x1_pro,transpose(z_pro)[delta,:];
28         label = "Proactive policy",
29         color = "dodgerblue4",
30         alpha = 0.50,
31     )
32     else
33     plot!(p1,x1_pro,transpose(z_pro)[delta,:];
34         label = nothing,
35         color = "dodgerblue4",
36         alpha = 0.50,
37     )
38     end
39 end
40 p1
41 end

```



### 3.2 Heuristics optimization

In order to implement an optimization routine, two objective (bivariate) functions are define for the proactive and reactive decision profiles, respectively. The same optimization algorithm and routine options used in past labs is used. The time limit was analyzed as follows for 10,000 simulations. A short, 60 second limit yield to optimal results considering the discrete nature of the problem. T shows the result for a 4 ft offset building optimization.

Policy	Variable	T = 25 years		T = 50 years	
		DR = 3%	DR = 7%	DR = 3%	DR = 7%
Proactive	Action [ft]	6.27	9.26	8.32	7.20
	$\Delta$ [ft]	1.0	0.52	6.05	2.74
	$\mu$ (NPV) [USM]	2.44	1.54	3.03	1.84
Reactive	Action [ft]	3.63	3.52	8.44	6.52
	$L_{max}$ [ft]	13.90	13.75	0.20	10.1
	$\mu$ (NPV) [USM]	2.47	1.57	3.36	2.10

Figure 6: Table 2

```

1 Random.seed!(2024)
2 options = Options(; time_limit = 60.0, f_tol_rel = 1e-3)
3 bounds = boxconstraints( lb = [0.0,0.0], ub = [14.0,14.0])
4 algorithm = ECA(; options = options)
5 result_pro = optimize(objective_function_Pro, bounds,algorithm)
6 result_re = optimize(objective_function_Re, bounds,algorithm)
7 print(result_pro)
8 print(result_re)

```

Optimization Result

=====

```

Iteration:      10
Minimum:        3.03727
Minimizer:      [8.37964, 4.77114]
Function calls: 140
Total time:     60.2470 s
Stop reason:    Maximum time exceeded.

```

Optimization Result

=====

```

Iteration:      11
Minimum:        3.03727
Minimizer:      [8.37964, 4.77114]
Function calls: 140
Total time:     61.4840 s
Stop reason:    Maximum time exceeded.

```

## 4 Conclusion

Sequential decision framework allows for add flexibility to, in this case, an elevation problem considering storm surges. The space of possibilities can grow quickly by incorporate uncertainty. Nevertheless, some of them can be handle by decision makers preferences, in particular, financial parameters such as the discount rate and the time horizon. Consider sea level rise adds complexity to the simulation given the number of realizations that are possible. The analysis shows that there are optimal decisions that minimize expected cashflows in the life span of the building. For a proactive profile, having a free-board distance to trigger an elevation action could yield in important “savings” in damages and losses due to hurricane surges. Another possible attitude, a reactive one, can also take the decision based on a threshold or flood level tolerance. As show in this work, the elevation action can take place in different moments of the time frame but using a Monte Carlo simulation is possible to get a rough mean estimate that allow for comparison between profiles and different state of the world. When compare to a static profile, that is, upfront decision making for elevation, the sequential decision can result in actions that are more cost-effective from a cost-benefit analysis perspective. F compares the results obtained in previous class labs (static upfront decision) with two scenarios. For a 50-year and 3% discount ratio, all models seem to indicate that an eight feet elevation should take place to satisfy free-boards, maximum year levels and minimize the NPV. When using a less attractive financial scenario, 50-year but 7% discount ratio, the upfront decision would be not to elevate the building. In contrast the adaptive policy presents a policy to reduce the expected NPV by having a 3 ft free-board  $\Delta$  and a 7 ft  $L_{max}$  for proactive and reactive

policies, respectively. Same results, yet more precise were obtained through heuristic optimization (see Table 2). The question on when to elevate is not directly answer, but a proactive policy yields better results than a reactive one, that for the example has to withstand one major flood event (7 ft flood) in order to act.

```

1  sows = let
2      N_SOW = 1000
3      sows = [SOW(rand(slr_scenarios),
4                  draw_surge_distribution(),
5                  draw_discount_rate("low")) for _ in 1:N_SOW]
6  end
7
8  mp = timeframe_50
9  function objective_function(a::Vector{Float64})
10     a = a[1]
11     a = Action((a)u"ft")
12     NPV = -[run_sim(a, s, mp) for s in sows] ./ 1e6
13     return mean(NPV)
14 end

```

objective\_function (generic function with 1 method)

```

1  let
2      elevations = 0.0:14.0
3      NPVs = [objective_function([e]) for e in elevations]
4      p = plot(elevations, NPVs;
5              xlabel = "Elevation action (ft)",
6              ylabel = "Mean NPV (US Millions)",
7              ylims = [1, trunc(maximum(NPVs)) + 1],
8              xlims = [0, 10],
9              label = "Static policy",
10             legend = :topright,
11             size = (500, 400),
12             color = "teal",
13             linewidth = 1,
14             markershape = :circle,
15             markercolor = "black",
16             markersize = 3)
17     plot!(p, x1_pro, z_pro[:, 4],
18         #label = "Proactive policy Δ = 3 ft",
19         label = "Proactive policy Δ = 4 ft",
20         color = "dodgerblue4",
21         markershape = :circle,
22         markercolor = "black",
23         linewidth = 1,
24         markersize = 3)
25     plot!(p, x1_re, z_re[:, 1],
26         #label = "Reactive policy Lmax = 7 ft",
27         label = "Reactive policy Lmax = 1 ft",

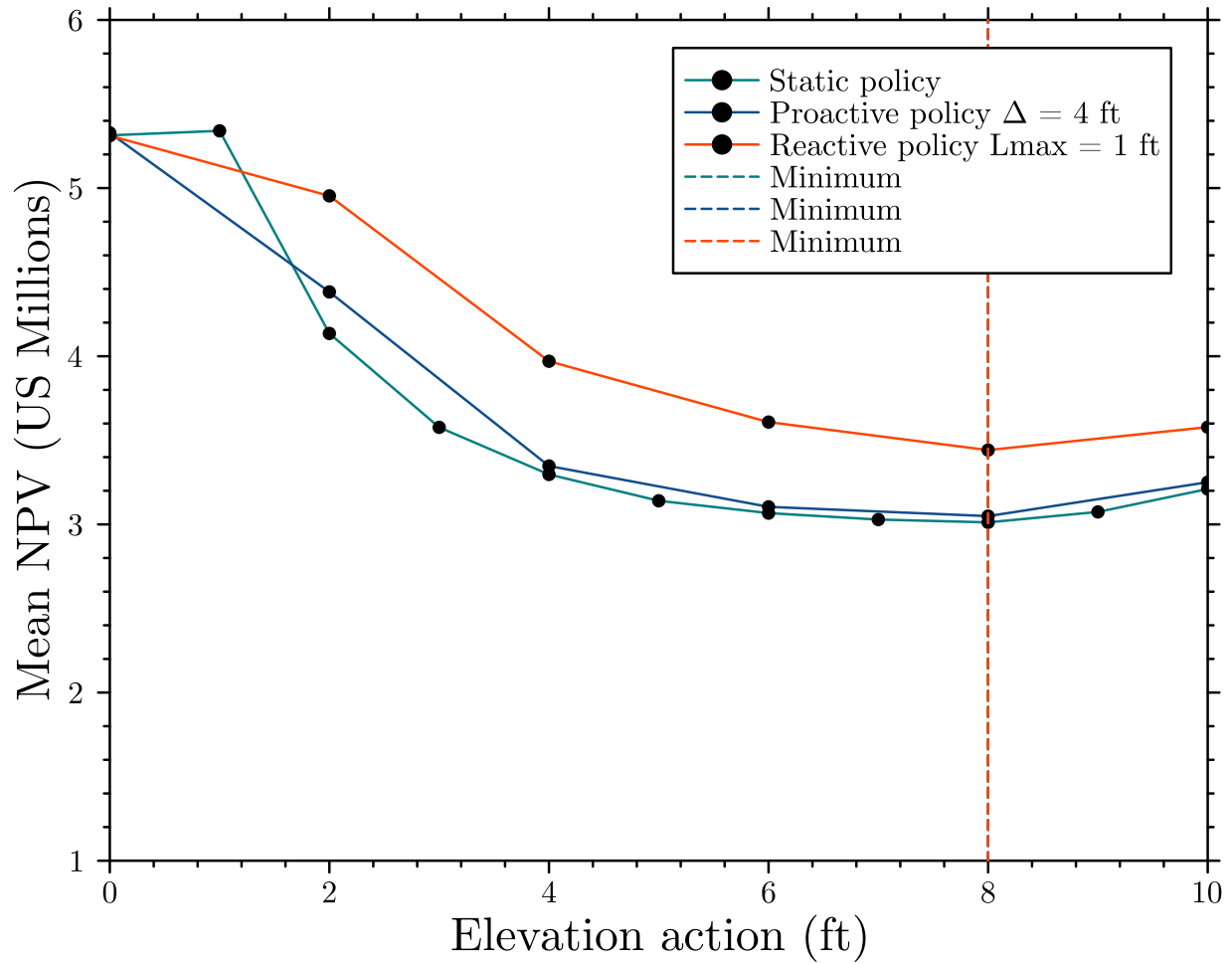
```



```

28     color = "orangered",
29     markershape = :circle,
30     markercolor = "black",
31     linewidth = 1,
32     markersize = 3)
33 vline!([findall(x-> x == minimum(NPVs), NPVs)][1]-[1]]; label = "Minimum",linewidth = 1, color = "dodgerblue4", style = :dash)
34 vline!([8]; label = "Minimum",linewidth = 1, color = "dodgerblue4", style = :dash)
35 vline!([8]; label = "Minimum",linewidth = 1, color = "orangered", style = :dash)
36 p
37 end

```



#### 4.1 Discussion and further work

Using exploratory modeling is very useful to incorporate many variables to simulation. Instead of working on the deterministic most realistic analysis, this methodology uses the uncertainties and different possible futures to inform and support decision making. In the context of climate risk management, this methodology and tools are more relevant than ever considering the extreme climate and its variations due to climate change. It also relevant considering the dynamics and assets and people at risk. In the case of elevation houses or buildings to minimize flooding and storm surge,

exploratory modeling, optimization, policy search, and scenario modeling are necessary to answer as much “what if?”, “how much?” and “when?” as possible considering several sources of uncertainty. The present work is just a small application of sequential or time dependent analysis where variables and their states change in time. For many cities, the sea level, the riverine behavior, the rain patterns, and other methodological variables are changing. Additionally, extremes are completely random in nature. Incorporating the quantitative measurements of intensities is relevant and in time. Therefore, forecasting allows to understand and simulate different scenarios and adaptive policies could inform different action sequences or parameters. The limitations of this work are in the number and quality of uncertainty management of the different variables: physical, financial, climate-related, as well as the assumptions in the model, for example, sampling one maximum level per year, neglecting hydrological behavior, event duration, damage progression, among others. Nevertheless, the present work helps to get closer to a policy that could define elevation heights and parameter that support those decisions, for example, following the sea level rise and trigger action if a given level is finally reached.

## 5 References

- Amini, Kooshan, and Jamie E. Padgett. 2023. “Probabilistic Risk Assessment of Hurricane-Induced Debris Impacts on Coastal Transportation Infrastructure.” *Reliability Engineering & System Safety* 240 (December): 109579. <https://doi.org/10.1016/j.ress.2023.109579>.
- Bernier, Carl, and Jamie E. Padgett. 2019. “Buckling of Aboveground Storage Tanks Subjected to Storm Surge and Wave Loads.” *Engineering Structures* 197 (October): 109388. <https://doi.org/10.1016/j.engstruct.2019.109388>.
- Buchanan, Maya K., Robert E. Kopp, Michael Oppenheimer, and Claudia Tebaldi. 2016. “Allowances for Evolving Coastal Flood Risk Under Uncertain Local Sea-Level Rise.” *Climatic Change* 137 (3-4): 347–62. <https://doi.org/10.1007/s10584-016-1664-7>.
- Davis, Darryl, Beth A. Faber, and Jerry R. Stedinger. 2008. “USACE Experience in Implementing Risk Analysis for Flood Damage Reduction Projects.” *Journal of Contemporary Water Research & Education* 140 (1): 3–14. <https://doi.org/10.1111/j.1936-704X.2008.00023.x>.
- FEMA. 2022. “Hazus 5.1 Flood Model Technical Manual.” *Department of Homeland Security Federal Emergency Management Agency*.
- Garner, Gregory G., and Klaus Keller. 2018. “Using Direct Policy Search to Identify Robust Strategies in Adapting to Uncertain Sea-Level Rise and Storm Surge.” *Environmental Modelling & Software* 107 (September): 96–104. <https://doi.org/10.1016/j.envsoft.2018.05.006>.
- Gori, Avantika, Ioannis Gidaris, James R. Elliott, Jamie Padgett, Kevin Loughran, Philip Bedient, Pranavesh Panakkal, and Andrew Juan. 2020. “Accessibility and Recovery Assessment of Houston’s Roadway Network Due to Fluvial Flooding During Hurricane Harvey.” *Natural Hazards Review* 21 (2): 04020005. [https://doi.org/10.1061/\(ASCE\)NH.1527-6996.0000355](https://doi.org/10.1061/(ASCE)NH.1527-6996.0000355).
- Oddo, Perry C., Ben S. Lee, Gregory G. Garner, Vivek Srikrishnan, Patrick M. Reed, Chris E. Forest, and Klaus Keller. 2020. “Deep Uncertainties in Sea-Level Rise and Storm Surge Projections: Implications for Coastal Flood Risk Management.” *Risk Analysis* 40 (1): 153–68. <https://doi.org/10.1111/risa.12888>.
- Wing, Oliver E. J., Nicholas Pinter, Paul D. Bates, and Carolyn Kousky. 2020. “New Insights into US Flood Vulnerability Revealed from Flood Insurance Big Data.” *Nature Communications* 11 (1): 1444. <https://doi.org/10.1038/s41467-020-15264-2>.