

Министерство науки и высшего образования Российской Федерации  
Санкт-Петербургский политехнический университет Петра Великого  
Инженерно-строительный институт  
Высшая школа гидротехнического и энергетического строительства

Работа допущена к защите

Директор ВШ ГиЭС

\_\_\_\_\_ Г.Л. Козинец

«\_\_\_» \_\_\_\_\_ 2020 г.

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Работа бакалавра

### **КОНСТРУИРОВАНИЕ И СТАТИЧЕСКИЙ РАСЧЕТ КОНТРФОРСНОЙ ПЛОТИНЫ С МАССИВНЫМИ ОГОЛОВКАМИ**

по направлению подготовки \_\_\_\_\_ 08.03.01 «Строительство»

Направленность (профиль) 08.03.01\_7 «Промышленное и гражданское  
строительство в сложных гидрогеологических условиях»

Выполнил

студент гр.з3130801/50701

Г.А. Юшманов

Руководитель  
доцент,  
к. т. н

И.Е. Фролова

Санкт-Петербург  
2020 г.

## **РЕФЕРАТ**

На 59 с., 19 рисунков, 1 таблица, 1 приложение.

**КЛЮЧЕВЫЕ СЛОВА:** ЧИСЛЕННЫЙ МЕТОД, КОНТРФОРСНАЯ ПЛОТИНА, ГИДРОТЕХНИЧЕСКОЕ СТРОИТЕЛЬСТВО, РЕЧНЫЕ ГИДРОТЕХНИЧЕСКИЕ СООРУЖЕНИЯ, PYTHON, NUMPY, MATPLOTLIB

В работе предложен численный метод расчета высоконапорных бетонных контрфорсных плотин с массивными оголовками на прочность и устойчивость. Данна общая информация о существующих типах гравитационных бетонных плотин. Написана программа для расчета элемента конструкции плотины и построения эпюров нормальных и касательных напряжений в сечении на языке Python с использованием среды разработки Jupyter Notebook и библиотек Numpy и Matplotlib.

## **ABSTRACT**

59 pages, 19 figures, 1 table, 1 appendix

**KEY WORDS:** NUMERICAL METHOD, BUTTRESS DAM, HYDROTECHNICAL CONSTRUCTION, RIVER HYDROTECHNICAL CONSTRUCTIONS, PYTHON, NUMPY, MATPLOTLIB

The study considers numerical calculation method of high-rise concrete buttress dams on durability and sustainability. Includes information about existing types of gravitational concrete dam constructions. Used self-developed program on Python language using Jupyter Notebook and libraries Numpy and Matplotlib.

# Содержание

<b>I Введение</b>	<b>3</b>
<b>II Бетонные плотины</b>	<b>3</b>
1 Классификация бетонных плотин	3
2 Гравитационные плотины	5
2.1 Массивные гравитационные плотины . . . . .	5
2.2 Контрфорсные плотины . . . . .	7
<b>III Анализ существующих контрфорсных плотин</b>	<b>8</b>
3 Обзор данных о плотинах	9
3.1 Дэниел-Джонсон, Канада . . . . .	9
3.2 Итайпу, Бразилия/Парагвай . . . . .	9
3.3 Пенсакола, США . . . . .	10
3.4 Плотина Зейской ГЭС, Россия . . . . .	10
4 Вывод	10
<b>IV Исходные данные для постановки исследований</b>	<b>10</b>
<b>V Назначения параметров контрфорсной плотины и В из условий требований устойчивости и прочности</b>	<b>11</b>
5 Геометрические характеристики расчетного сечения	12
6 Нагрузки	13
6.1 Нагрузка от собственного веса плотины . . . . .	14
6.2 Гидростатическая нагрузка . . . . .	16
6.3 Противодавление . . . . .	16
6.4 Давление наносов . . . . .	17
6.5 Волновая нагрузка . . . . .	17
6.6 Давление льда . . . . .	18

6.7 Сочетания нагрузок . . . . .	18
<b>7 Нахождение оптимальных параметров <math>n</math> и <math>B</math> численным методом</b>	<b>18</b>
7.1 Общее описание метода . . . . .	19
7.2 Функция статического расчета секции плотины . . . . .	20
7.3 Функция построения графика зависимости $B$ от $n$ . . . . .	21
7.4 Функция построения графика зависимости $B$ и $n$ от высоты профиля плотины . . . . .	22
7.5 Графики зависимости $B$ от $n$ . . . . .	22
<b>VI Сопоставление касательных напряжений по подошве контрфорсной плотины с предельными значениями</b>	<b>23</b>
<b>8 Метод расчета</b>	<b>24</b>
<b>9 Вычисление напряжений</b>	<b>24</b>
9.1 Нормальные напряжения $\sigma_z$ . . . . .	24
9.2 Нормальные напряжения, действующие на ортогональных площадках $\sigma_x$ . . . . .	25
9.3 Касательные напряжения $\tau_{xz}$ . . . . .	26
9.4 Главные нормальные напряжения $\sigma_{1,2}$ . . . . .	27
<b>10 Эпюры напряжений в секции плотины</b>	<b>28</b>
10.1 Для высоты плотины 60 метров . . . . .	28
10.2 Для высоты плотины 80 метров . . . . .	29
10.3 Для высоты плотины 100 метров . . . . .	30
10.4 Рекомендации по назначению марки бетона для плотины .	31
<b>VII Рекомендации по назначению параметров контрфорсной плотины для различных её высот</b>	<b>31</b>

# **Часть I**

## **Введение**

Для комплексного использования водных ресурсов рек или борьбы с разрушительным действием водной стихии возводятся гидроузлы, в состав которых входят гидротехнические сооружения различного назначения. Гидротехнические сооружения разделяются на общие и специальные. К специальным гидротехническим сооружениям относятся сооружения, возводимые для какой-либо одной отрасли водного хозяйства, например гидроэнергетики, водного транспорта, инженерной мелиорации, водоснабжения, обводнения и т.д. Общие гидротехнические сооружения входят практически во все гидроузлы. К ним относятся такие сооружения как водоподпорные, водопроводящие, водозaborные и водосбросные.

В данной работе из всего многообразия общих гидротехнических сооружений рассматриваются только подпорные сооружения. Целью возведения водоподпорных сооружений – глухих плотин является поднятие уровня воды в реке на некоторую высоту, создание перепада уровней перед плотиной и за ней и сосредоточенного напора, а также организация водохранилища. По использованию создаваемого напора глухой плотиной различают гидроузлы: высоконапорные водохранилищные, низконапорные и деривационные. В современных условиях подпорные плотины выполняются из бетона или из грунтовых материалов. В дальнейшем в работе будут рассматриваться водоподпорные сооружения - плотины высоконапорных водохранилищных гидроузлов выполненных из бетона.

# **Часть II**

## **Бетонные плотины**

### **1 Классификация бетонных плотин**

Бетон как строительный материал был известен еще со времен Древнего Рима. Широкое использование его в гидротехническом строительстве началось во второй половине XIX века. Этот материал отличается не только прочностью, но и морозостойкостью и водонепроницаемостью, что было доказано на практике. Бетонные плотины можно классифицировать по различным признакам. По эксплуатационному назначению выделяют:

- Глухие или водоподпорные участки, служащие для создания разности уровней воды (напора). Через них вода не переливается из водохранилища в нижний бьеф (Рис.1а);
- Водосливные участки, служащие для холостого сброса воды, регулирования уровня воды в верхнем бьефе и защиты от переливания воды через гребень глухой плотины в период паводка (Рис.1б);
- Участки с напорными водоводами, служащие для подведения воды под напором к гидротурбинам или насосам (Рис.1в);

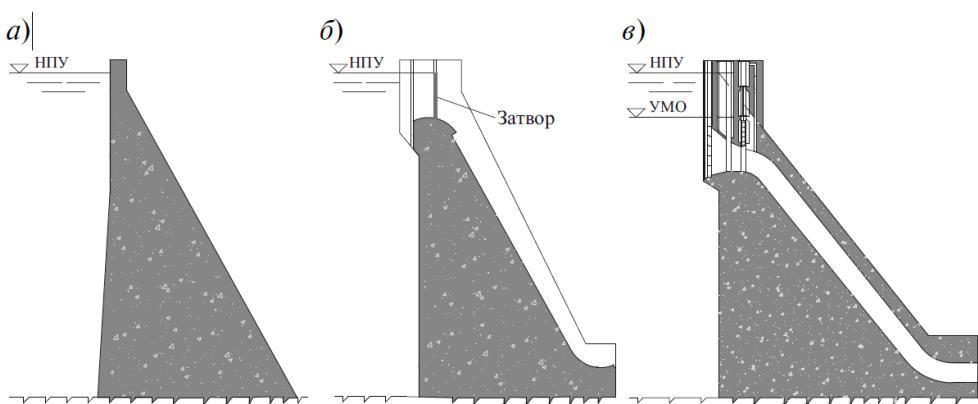


Рис.1  
а-глухая плотина; б-водосливная часть; в-станционная часть

По принципу сопротивления сдвигающим силам плотины делятся на:

- Гравитационные
- Арочные

Гравитационные плотины сопротивляются сдвигу за счет сил трения возникающих на контакте между основанием плотины и подстилающей поверхностью. Силы, действующие на плотину пропорциональны весу плотины и давлению воды, пригружающей плотину со стороны верхнего и нижнего бьефов, за вычетом сил противодавления, возникающих на контакте плотины с основанием из-за проникновения некоторой части воды под плотину.

Все существующие типы конструкций гравитационных плотин возводятся отдельными секциями, которые самостоятельно воспринимают действующую на плотину нагрузку. По сути гравитационная плотина

разрезается на секции вертикальными перпендикулярными гребнями плотины плоскостями. Промежутки между секциями называются деформационными швами. Они уплотняются специальными устройствами, предотвращающими свободное вытекание воды из верхнего бьефа в нижний, они же объединяют всю конструкцию плотины в единое напорное сооружение. Данное мероприятие необходимо для предотвращения образования трещин, которые могут появиться из-за неравномерных осадок грунта под воздействием веса плотины и других различных нагрузок, а также сезонных перепадов температур. Ширина секций плотины обычно составляет 10-12 метров.

Арочные плотины сопротивляются сдвигу за счет упора в берега, а вес играет незначительную роль. Арочные плотины разделяют на блоки только во время возведения с целью ослабить влияние тепловыделения бетона, а затем швы цементируют.

## 2 Гравитационные плотины

По конструктивным признакам в свою очередь гравитационные плотины делятся массивные и контрфорсные.

### 2.1 Массивные гравитационные плотины

Основными нагрузками на массивную гравитационную плотину являются ее собственный вес и давление воды со стороны верхнего бьефа. В связи с этим поперечный профиль плотины выполняется треугольной формы с дополнением конструктивных элементов (рис.2). Вершина профиля располагается на отметке нормального подпорного уровня (НПУ) или иногда на отметки форсированного подпорного уровня (ФПУ). Форма верховой грани плотины выполняется в зависимости от грунтов основания - при сильных скальных грунтах её можно выполнить вертикальной для уменьшения расхода бетона, для менее прочной скалы слабонаклонной.

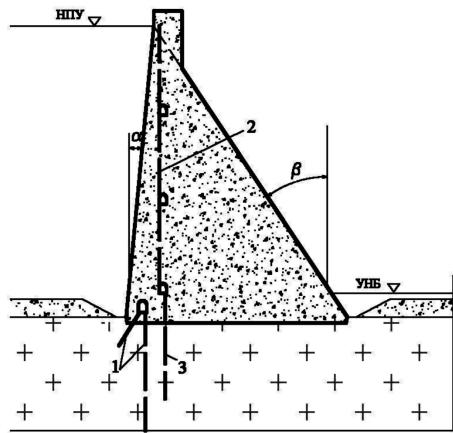


Рис.2

Схема профиля массивной гравитационной плотины

Наклон граней характеризуется коэффициентами заложения  $n$  и  $m$  ( $n$ -верховая грань;  $m$ -низовая грань), что численно равняется тангенсам углов наклона грани к вертикалам. Уклоны верховой и низовой граней плотины обозначаются тангенсами углов наклона к вертикалам. Ширина плотины по основанию определяется суммой коэффициентов заложения, умноженной на высоту треугольного профиля:

$$B = H_t(m + n)$$

Гребень плотины назначается конструктивно. Как правило, его делают такой ширины, чтобы по нему можно было соорудить автомобильную или железную дорогу, в пределах 6-10 метров. Если проезд не планируется, гребень имеет ширину не менее 2 метров для безопасного прохода. При устройстве проезда, остальные конструкции гидроузла должны выдерживать нагрузки от транспорта, то есть их проектируют как опоры мостов.

Для предотвращения переливания воды через гребень плотины нужно учесть нагон и накат ветровых волн, и сделать гребень выше уровня воды на величину, определяемую суммой высота наката волн  $h_{nt}$ , возышения среднестатистической волны над уровнем воды, называемого нагоном волны  $h_w$  и дополнительного запаса надежности  $a$ .

$$d_c = h_w + h_{nt} + a$$

При назначении отметки гребня плотины следует учесть два расчетных варианта:

- Отметка НПУ + многолетняя высота волны обеспеченностью 2 процента для 1 и 2 классов сооружения и 4 процента для 3 и 4 классов;

- Отметка ФПУ + повторяющаяся раз в 5 лет волна меньшей высоты;

Подошва бетонной плотины устраивается всегда на отметки прочной скалы после удаления аллювиальных отложений и слабого поверхностного слоя, как наиболее трещиноватого. Иногда под плотиной основание дополнительно укрепляют с помощью цементации на небольшую глубину.

Для снижения давления фильтрационной воды на подошву, уменьшения градиента напора и расхода фильтрационного потока через трещины в основании у верхней грани плотины устраивается более глубокая цементация, глубина которой определяется геологическими исследованиями. Для устройства цементации в скальном основании бурятся скважины диаметром 10-15 см., в которые нагнетается цементационный раствор. За цементационной завесой в основании устраивается дренаж для отвода воды, профильтровавшей через цементационную завесу и в обход ее.

В теле плотины у верховой ее грани тоже устраивается дренаж, назначение которого перехватить фильтрационную воду и не дать ей распространить в глубину плотины. Дренаж тела плотины представляет собой систему вертикальных дренажных скважин диаметром 10-30 см. и продольных галерей. Профильтровавшая вода стекает через выпуск под уровень нижнего бьефа. Галереи используются не только для сбора фильтрационной воды, но и для контроля за состоянием тела плотины. В них устанавливается контрольно-измерительная аппаратура (КИА).

## 2.2 Контрфорсные плотины

По конструктивным признакам контрфорсные плотины делятся на:

- с массивными оголовками;
- с плоскими перекрытиями;
- со сводчатыми перекрытиями;

Плотины с массивными оголовками это отдельно стоящие вертикальные стены-контрфорсы с уширениями до смыкания со стороны верхнего бьефа (Рис.3).

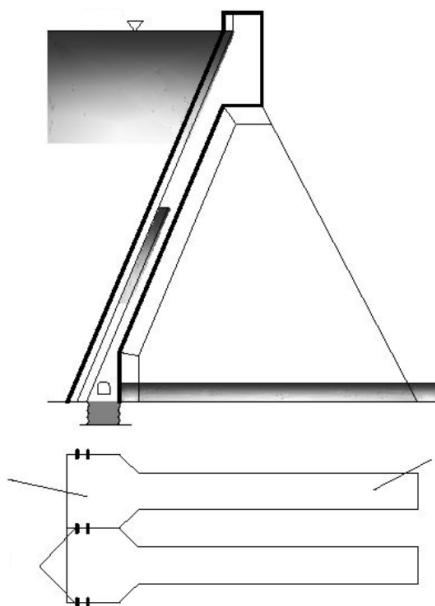


Рис.3

Контрфорсная плотина с массивными оголовками: 1-оголовок;  
2-контрфорс; 3-уплотнение

Массивно-контрфорсные плотины наиболее распространенные из-за простоты конструкции. Напорная грань у них как правило плоская.

Характерные размеры этого типа плотины: ширина секции (оголовка) 11-20м.; толщина контрфорса 4-8м.; наклон верховой грани  $n=0,40-0,45$ .

Конструкция и размеры гребня контрфорсной плотины определяются теми же условиями, что и у массивной гравитационной бетонной плотины. Сопряжение плотины со скальным основанием осуществляется с помощью устройства патерны в нижней её части, из которой производится цементация основания завесы.

Проведенный обзор конструкций бетонных подпорных плотин позволил выбрать в качестве дальнейшего исследования контрфорсную плотину с массивными оголовками.

# Часть III

## Анализ существующих контрфорсных плотин

### 3 Обзор данных о плотинах

#### 3.1 Дэниел-Джонсон, Канада



Река - Маникуаган, Высота - 214 метров, Длина - 1314 метров, Объем бетона - 2.2 млн м<sup>3</sup>, 14 контрфорсов, Толщина в самой тонкой точке - 22.5 метра, Ширина гребня - 3 метра, Верховая грань покрыта асфальтом, Самая крупная арка - 160 метров, Водохранилище кольцеобразной формы, Площадь водохранилища - 1942 км<sup>2</sup>

#### 3.2 Итайпу, Бразилия/Парагвай



Река - Парана, Общая длина (включая каменно-набросную часть) - 7.2 км, Высота плотины - 196 метров, Площадь водохранилища - 1460 км<sup>2</sup>, Объем бетона - 15,57 млн м<sup>3</sup>

### **3.3 Пенсакола, США**



Высота - 46 метров, Общая длина - 2 км, Длина подпорного участка - 1.3 км, Ширина арки - 18 метров, Ширина контрфорса - 7.8 метра

### **3.4 Плотина Зейской ГЭС, Россия**



Высота плотины - 115 метров, Уклон верховой грани - 0.15, Уклон низовой грани - 0.8, Длина плотины - 714 метров, Ширина контрфорса - 7 метров, Ширина секции - 15 метров

## **4 Вывод**

На основании проведенного обзора для дальнейших исследований выбрана контрфорсная плотина с массивным оголовком с размерами по высоте от 60 до 100 метров с шагом 10 метров, отношением ширины оголовка к ширине контрфорса 0.5 и для следующих характеристик грунта основания

## Часть IV

# Исходные данные для постановки исследований

Контрфорсные плотины лучше строить в районах с положительной температурой. Для исследования взят подобный район, следовательно при составлении сочетаний нагрузок можно не учитывать нагрузку от давления льда.

Характеристики грунта основания:

- $f = 0.75$
- $c = 20$  т

Геометрические характеристики профиля плотины:

- Высота треугольного профиля  $H_t = 60$  м
- Уклон верховой грани  $n = 0.4$
- Ширина плотины по основанию  $B = 60$  м

Геометрические характеристики одной секции плотины:

- Ширина оголовка  $D = 20$  м
- Ширина контрфорса  $d = 0.5D = 10$  м
- Толщина оголовка  $b = 0.15H_t$

Глубина в верхнем бьефе - 60 метров

В нижнем - 7.2 метра

Высота волны в водохранилище - 0.8 метра

Толщина слоя наносов - 24 метра

# Часть V

## Назначения параметров контрфорсной плотины на В из условий требований устойчивости и прочности

Гравитационные плотины сопротивляются нагрузкам за счет собственной прочности и прочности контакта с основанием. Прочность достигается подбором такой формы конструкции секции, чтобы в ней отсутствовали растягивающие напряжения, а также зоны, в которых касательные напряжения достигают предельных значений.

### 5 Геометрические характеристики расчетного сечения

Для выполнения статического расчета необходимо найти центр тяжести расчетного сечения. Положение центра тяжести расчетного сечения:

$$x_A = (0.5 * B^2 d + ab^2 + a^2(b + a\sqrt{1 + n^2}/3)\sqrt{1 + n^2})/F$$

$$x_B = B - x_A$$

Момент инерции расчетного сечения в главной центральной системе координат (формула для простоты разделена на 6 слагаемых):

$$Jy_0 = B^3 d / 12$$

$$Jy_1 = ab^3 / 6$$

$$Jy_2 = \frac{a^4}{18}(1 + n^2)^{1.5}$$

$$Jy_3 = (x_A - 0.5B)^2 Bd$$

$$Jy_4 = 2(x_A - 0.5b)^2 ab$$

$$Jy_5 = a^2(x_A - b - a\sqrt{1+n^2}/3)^2\sqrt{1+n^2}$$

$$Jy = \sum J_{yi}$$

Моменты сопротивления

$$W_A = \frac{J_y}{|x_A|}$$

$$W_B = \frac{J_y}{|x_B|}$$

$$F = Bd + a(2b + a\sqrt{1+n^2})$$

$$a = (D - d)/2$$

## 6 Нагрузки

Список нагрузок, действующих на секцию плотины имеет следующий вид:

- Собственный вес плотины
- Гидростатическая нагрузка
- Противодавление
- Давление взвешенных наносов
- Волновая нагрузка

Необходимо найти сумму сил от этих нагрузок и их моментов, затем используя критерий отсутствия растягивающих напряжений и критерий устойчивости плотины найти оптимальные параметры профиля плотины - ширину по основанию и уклон верховой грани.

Для простоты большие математические выражения были разделены на составные части.

## 6.1 Нагрузка от собственного веса плотины

При определении собственного веса плотины самым простым и понятным путем будет разделение бокового профиля сегмента плотины на элементы.

Вес элемента контрфорса с треугольным профилем:

$$G_1 = 0.5\gamma_c BH_t d$$

где  $\gamma_c$  - удельный вес бетона.

Момент веса этого элемента относительно главной центральной оси у:

$$M_1 = \gamma_c H_t Bd(B - 3x_A + nH_t)/6$$

Вес элементов оголовка контрфорса:

$$G_2 = \gamma_c H_t (a\sqrt{1+n^2} + 2b)a$$

Момент веса этого элемента относительно главной центральной оси у (выражение состоит из нескольких частей):

$$M_{2,1} = b(2x_A - b - nH_t)$$

$$M_{2,2} = a\sqrt{1+n^2}(x_A - b - (nH_t)/2 - (a\sqrt{1+n^2})/3)$$

$$M_2 = -\gamma_c H_t a(M_{2,1} + M_{2,2})$$

Вес элемента гребневого оголовка с прямоугольным профилем и его момент относительно главной центральной оси у:

$$G_3 = \gamma_c B_c D d_c$$

$$M_3 = -G_3(x_A - nH_t - e_c)$$

где:  $d_c = hw + d_{hw} + 0.6$  - возвышение гребня плотины над уровнем верхнего бьефа,

$d_{hw} = \pi h_w^2 / L_w$  - высота наката волны,  $B_c$  - ширина гребня плотины.

Вес элемента гребневого оголовка с треугольным профилем и его момент относительно главной центральной оси у:

$$G4 = 0.5\gamma_c Da_1^2/n$$

$$M4 = -G4(x_A - nH_t + 2a_1/3)$$

где:  $a_1 = (0.5B_c - e_c)$ ,  $a_2 = (0.5B_c + e_c)$

Вес изымаемого элемента треугольного сечения:

$$G_5 = -0.5\gamma_c n D a_2^2 / m^2$$

Момент веса этого элемента относительно главной центральной оси у (выражение состоит из нескольких частей):

$$M_{5,1} = \frac{2na_2}{3m}$$

$$M_5 = -G_5(x_A - b - nH_t + M_{5,1})$$

Для нахождения веса изымаемого элемента трапецидального сечения  $G_6$  необходимо найти следующие величины:

$$\begin{aligned} a^* &= \frac{a_2(b + a\sqrt{1 + n^2})\sqrt{1 + n^2}}{mb} \\ b^* &= x_A - b - nH_t - a\sqrt{1 + n^2} \\ n_1 &= \frac{1 - nm}{n + m} \end{aligned}$$

Вес изымаемого элемента трапецидального сечения (выражение состоит из нескольких частей):

$$G_{6,1} = a^* - \frac{2a\sqrt{1 + n^2}H_t}{3B}$$

$$G_6 = -\gamma_c a^2 G_{6,1}$$

Момент веса этого фрагмента (выражение состоит из нескольких частей):

$$\begin{aligned} M_{6,1} &= \frac{n(2 + n^2)}{\sqrt{1 + n^2}} + \frac{n_1(2 - n^2)}{\sqrt{1 + n_1^2}} \\ M_{6,2} &= nn_1a^* + (n + n_1)b^* + \frac{3aM_{6,1}}{8} \\ M_{6,2,2} &= b^* + \frac{na}{2} + \frac{2a}{3\sqrt{1 + n^2}} \\ M_{6,3} &= a^*M_{6,2,2} - 2aM_{6,2}/3 \\ M_6 &= \gamma_c a^2 M_{6,3} \end{aligned}$$

## 6.2 Гидростатическая нагрузка

Гидростатическую нагрузку в данной работе предлагается разделить на вертикальную и горизонтальную составляющую.

Нагрузка со стороны верхнего бьефа, горизонтальная составляющая:

$$W_{up,hrz} = 0.5\gamma_w H_t^2 D$$

$$M_{up,hrz} = \gamma_w H_t^3 D / 6$$

где  $\gamma_w$  - удельный вес воды.

Вертикальная составляющая:

$$W_{up,vrt} = nW_{up,hrz}$$

$$M_{up,vrt} = -\gamma_w n D H_t^2 (3x_A - nH_t) / 6$$

Нагрузка со стороны нижнего бьефа, горизонтальная составляющая:

$$W_{low,hrz} = -0.5\gamma_w H_2^2 D$$

$$M_{low,hrz} = W_{low,hrz} * H_2 / 3$$

где  $H_2$  - глубина воды в нижнем бьефе.

Вертикальная составляющая:

$$W_{low,vrt} = (n - \frac{Bd}{H_t D}) * W_{low,hrz}$$

Выражения для момента состоит из нескольких частей:

$$M_{low,vrt,1} = |x_A| - b - \frac{a}{2} - \frac{nH_2}{3}$$

$$M_{low,vrt,2} = x_B - mH_2$$

$$M_{low,vrt,3} = 2naM_{low,vrt,1} + mM_{low,vrt,2}d$$

$$M_{low,vrt} = \frac{M_{low,vrt,3}W_{low,vrt}}{D}$$

## 6.3 Противодавление

Нагрузка от противодавления действует на подошву плотины.

Сила взвешивания:

$$U_1 = -\gamma_w H_2 [Db + \frac{a(D+d)}{2} + (B-b-a)d]$$

$$M_{U1} = 0$$

Фильтрационная составляющая, передающаяся на подошву плотины через бетонную подушку:

$$U_2 = -\gamma_w(H_t - H_2)(b + a)D$$

Момент этой силы:

$$M_{U2} = -U_2(|x_A| - \frac{b + a}{2})$$

## 6.4 Давление наносов

Данную нагрузку также предлагается разбить на горизонтальную и вертикальную составляющие.

Горизонтальная составляющая:

$$W_{allu,hrz} = 0.5\gamma_{allu}H_{allu}^2D$$

Момент:

$$M_{allu,hrz} = W_{allu,hrz}H_{allu}/3$$

где  $\gamma_{allu}$  - удельный вес взвешенных уплотнившихся наносов,  $H_{allu}$  - толщина слоя наносов,

Вертикальная составляющая:

$$W_{allu,vrt} = 0.5\gamma_{allu}nH_{allu}^2D$$

Момент:

$$M_{allu,vrt} = -W_{allu,vrt}(|x_A| - \frac{nH_{allu}}{3})$$

## 6.5 Волновая нагрузка

Волновая нагрузка, очевидно, действует только в то время, когда поверхность водохранилища не покрыта льдом, за исключением теплых районов.

Горизонтальная составляющая волновой нагрузки:

$$W_{wave,hrz} = \gamma_w h_w (\frac{L_w}{\pi} + \frac{h_w}{2}) \frac{D}{2}$$

где  $L_w$  - длина волны,

Момент этой нагрузки:

$$M_{wave,hrz} = W_{wave,hrz}(H_t - \frac{L_w}{2\pi} + \frac{3h_w}{8})$$

Вертикальная составляющая:

$$W_{wave,vrt} = W_{wave,hrz}n$$

Моментом вертикальной составляющей можно пренебречь.

## 6.6 Давление льда

Поскольку в данной работе выбран район строительства с положительными температурами, следующие выражения не используются. При проектировании в холодных районах следует учесть эту составляющую при составлении сочетаний нагрузок.

$$W_{ice} = R_{ice}h_{ice}D$$

$$M_{ice} = W_{ice}(H_t - 0.45h_{ice})$$

## 6.7 Сочетания нагрузок

Для строительного периода:

$$N = G_1 + G_2 + G_3 + G_4 + G_5 + G_6$$

$$Q = 0$$

$$M = M_1 + M_2 + M_3 + M_4 + M_5 + M_6$$

Для эксплуатационного периода:

$$N = G_1 + G_2 + G_3 + G_4 + G_5 + G_6 + W_{up,vrt} + W_{low,vrt} + U_1 + U_2 + W_{wave,vrt} + W_{allu,vrt}$$

$$Q = W_{up,hrz} + W_{low,hrz} + W_{allu,hrz} + W_{wave,hrz}$$

$$M_{water} = M_{up,hrz} + M_{up,vrt} + M_{low,hrz} + M_{low,vrt}$$

$$M = M_1 + M_2 + M_3 + M_4 + M_5 + M_6 + M_{water} + M_{U2} + M_{allu,hrz} + M_{allu,vrt} + M_{wave,hrz}$$

## 7 Нахождение оптимальных параметров $n$ и $B$ численным методом

Параметры  $n$  и  $B$  находятся из условий:

- Отсутствие во всех точках растягивающих напряжений
- Устойчивость плотины сдвигающим усилиям

- При всех вышеперечисленных - минимальный расход бетона

Условие отсутствия растягивающих напряжений выражается уравнением:

$$[\sigma_{z,A}]B^2 - NB + 6M = 0$$

Параметр  $[\sigma_{z,A}]$  - допускаемое напряжение в точке А, совмещающий в себе условие прочности и экономичности. Подбирать его следует, начиная с минимального значения 10 тонн на квадратный метр. Если этого оказывается недостаточно для обеспечения отсутствия растягивающих напряжений, можно увеличивать этот параметр, вплоть до 30.

Условие устойчивости секции плотины имеет следующий вид:

$$fN + cB - k_3Q = 0$$

где  $f, c$  - характеристики грунта основания,  
 $k_3$  - коэффициент запаса в зависимости от класса надежности сооружения. Для высоконапорных плотин (от 60 метров высотой) следует принять 1,25.

Для каждого значения  $n$  существует такое значение  $B$ , при котором выполняются оба условия.

В данной работе предлагается найти данные параметры численным методом.

## 7.1 Общее описание метода

Программа, предлагаемая в данной работе состоит из нескольких частей:

- Блок импорта необходимых для работы библиотек,
- Функция расчета нагрузок, критериев устойчивости, прочности и построения эпюров, далее - функция статического расчета,
- Функция построения графика зависимости  $B(n)$  с учетом критериев прочности и устойчивости,
- Функция построения графиков зависимости параметров  $n$  и  $B$  от высоты плотины,

В программировании функции используются для тех участков кода, которые повторяются несколько раз с целью избежать их повторения. Функция может иметь свои внутренние переменные, выполнять над ними различные операции.

Помимо внутренних переменных в функцию можно передать переменные из основного блока программы, такие переменные называются параметрами функции.

Функция также может возвращать в основной блок программы переменные.

Функция статического расчета принимает на вход несколько параметров, самые важные из них - уклон верховой грани  $n$ , ширину секции плотины по основанию  $B$ , высоту треугольного профиля плотины  $H_t$

Функция, используя выражения для вычисления нагрузок, может по усмотрению пользователя вернуть значения критериев устойчивости и отсутствия растягивающих напряжений, значения напряжений, а также проверить, не превышают ли касательные напряжения во всех точках предельных значений.

Критерии отсутствия растягивающих напряжений и устойчивости принимают вид:

$$C_{nostretch} = [\sigma_{z,A}]B^2 - NB + 6M$$

$$C_{sustain} = fN + cB - k_3Q$$

Компьютер перебирает все возможные значения  $n$  и  $B$  в заданном диапазоне. Те значения  $n$  и  $B$ , при которых  $C_{nostretch}$  и  $C_{sustain}$  максимально близки к нулю записываются в массив.

На основе полученных значений строятся два графика, один для условия отсутствия растягивающих напряжений, другой для условия устойчивости. Если все выполнено правильно, графики должны пересекаться в одной точке, которая и будет оптимальным решением.

## 7.2 Функция статического расчета секции плотины

Общий алгоритм работы функции:

- Принять на вход в качестве параметров геометрические характеристики плотины:  $B$ ,  $n$ ,  $H_t$ ,  $D$ ,  $d$ , допускаемое напряжение  $\sigma_z$  в точке А
- Найти значения всех нагрузок всех видов
- Собрать значения нагрузок в сочетания
- На основании сочетаний вычислить критерии отсутствия растягивающих напряжений и устойчивости

- Вернуть эти значения в основной блок программы
- Вычислить нормальные и касательные напряжения по всем точкам вдоль оси  $x$
- Проверить, не превышают ли напряжения предельных значений
- Построить графики напряжений

### 7.3 Функция построения графика зависимости $B$ от $n$

Цель функции - построить график зависимости ширины профиля плотины по основанию от уклона верховой грани при условии соблюдения критериев отсутствия растягивающих напряжений и устойчивости.

По графику необходимо найти точку пересечения двух линий и записать соответствующие ей значения  $B$  и  $n$ .

В будущих версиях программы предлагается находить решение численным методом из системы уравнений критериев прочности и устойчивости.

Общий алгоритм работы функции:

- Получить на входе данные о высоте треугольного профиля плотины, для которого выполняется расчет
- Перебрать  $B \times n$  комбинаций для каждой пары  $B$  и  $n$
- Для каждой пары найти те, при которых выполняются критерии прочности и устойчивости
- Записать эти пары в массив
- Вывести пары значений в виде графиков, отдельно для каждого критерия

Если все сделано правильно, график по критерию устойчивости - убывающий, по критерию отсутствия растягивающих напряжений - возрастающий.

Точка пересечения этих графиков будет оптимальным решением, удовлетворяющим всем критериям.

## 7.4 Функция построения графика зависимости В и n от высоты профиля плотины

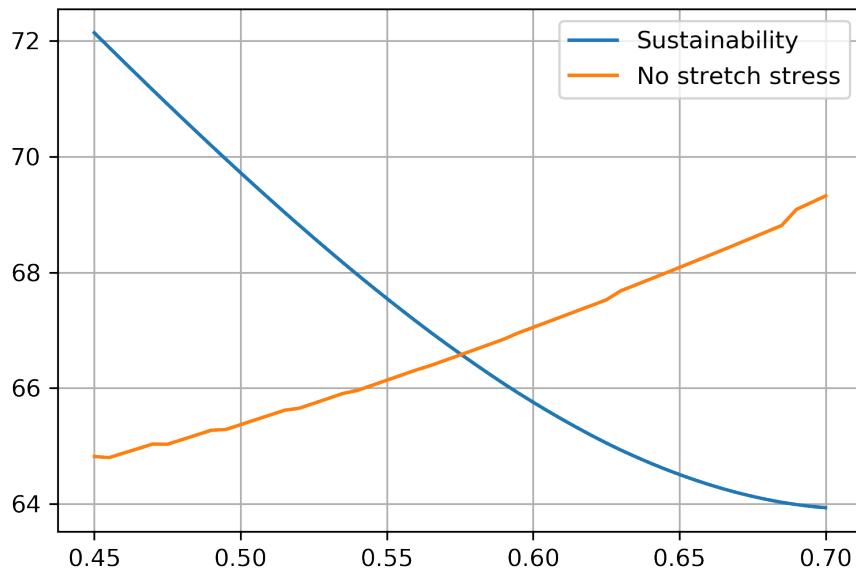
Цель данной функции - визуализировать, как найденные значения В и n соотносятся при разных высотах плотины.

Функция содержит в себе полученные ранее оптимальные значения для разных высот в виде массивов. На основании этих данных строятся два графика - зависимости  $B(H_t)$  и  $n(H_t)$ .

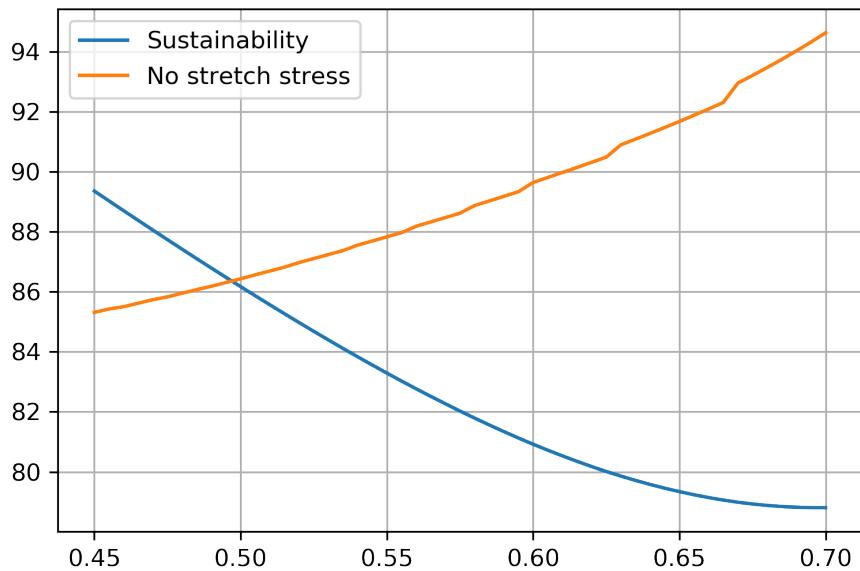
Поскольку графики в данных зависимостях состоят только из трех точек, а на графике  $n(H_t)$  это особенно заметно, выполнено сглаживание последнего с помощью функции `make_interp_spline` из библиотеки SciPy.

## 7.5 Графики зависимости В от n

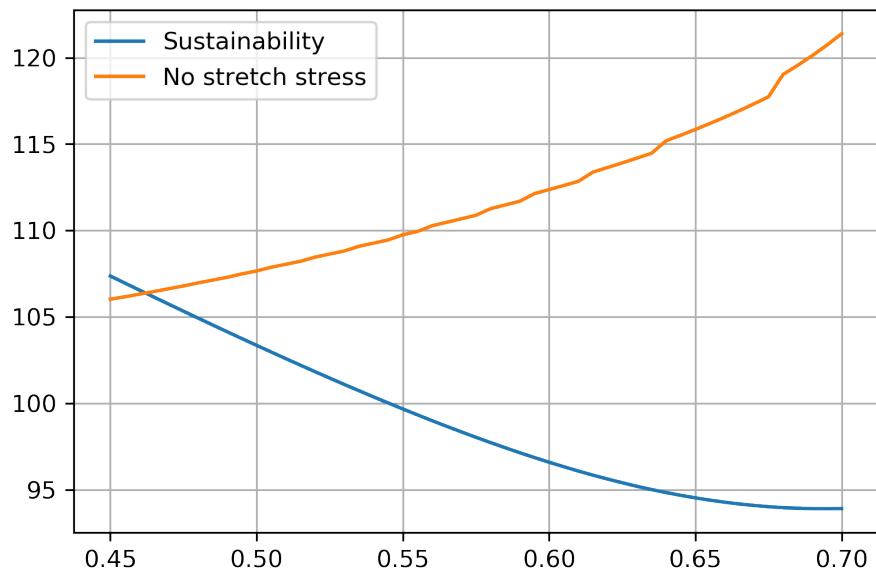
$$H_t = 60$$



$$H_t = 80$$



$H_t = 100$



# Часть VI

## Сопоставление касательных напряжений по подошве контрфорсной плотины с предельными значениями

### 8 Метод расчета

Используется библиотека **NumPy**, работающая массивами чисел.

Создается массив с координатами  $x \in [x_A, x_B]$ . При подстановке массива в выражение, вычисления производятся для каждого элемента массива, на выходе, получается массив такого же размера.

Далее строится график с использованием библиотеки **Matplotlib**, где за ось x берется массив с координатами, за ось y - массив полученных значений.

### 9 Вычисление напряжений

#### 9.1 Нормальные напряжения $\sigma_z$

Нормальные напряжения  $\sigma_z$  допускается определять по формуле внецентренного сжатия:

$$\sigma_z = \frac{N}{F} + \frac{M}{J_y}x$$
$$x \in [x_A, x_B]$$

За начало координат принимается центр тяжести расчетного сечения, точки слева имеют знак минус, справа - плюс.

Из формулы можно найти  $\sigma_z$  в точках А, В:

$$\sigma_{z,A} = \frac{N}{F} + \frac{M}{J_y}x_A$$
$$\sigma_{z,B} = \frac{N}{F} + \frac{M}{J_y}x_B$$

Найденные напряжения не могут быть отрицательными, в противном случае в бетоне возникнут зоны растяжения.

Условие экономичности плотины - напряжение  $\sigma_{z,A}$  близко к нулю.

По полученным значениям строится эпюра напряжений  $\sigma_z$  для визуализации полученных данных и их проверки. Эпюра принимается линейной.

## 9.2 Нормальные напряжения, действующие на ортогональных площадках $\sigma_x$

Для нахождения нормальных напряжений  $\sigma_x$  потребуется сначала найти давление воды и наносов на грани плотины:

$$p_A = \gamma_w H_t + \gamma_{allu} H_{allu}$$

$$p_B = \gamma_w H_2$$

Со стороны низовой грани действует только гидростатическое давление.

Напряжения  $\sigma_x$  определяются граничными условиями - в точках А и В:

$$\begin{aligned}\sigma_{x,A} &= (1 - n^2)p_A + n^2\sigma_{z,A} \\ \sigma_{x,B} &= (1 - m^2)p_B + m^2\sigma_{z,B}\end{aligned}$$

Эпюра строится на основе следующего выражения:

$$\sigma_x = \frac{\sigma_{x,A}(x_B - x)D + \sigma_{x,B}(x - x_A)d}{B[d(x)]}$$

Для каждой точки на оси х находится напряжение  $\sigma_x$ , на основании полученного массива значений строится эпюра. Величина  $[d(x)]$  зависит от текущей координаты x:

$$[d(x)] = \begin{cases} D, & x \in [x_A; x_{C1}] \\ D - \frac{D-d}{x_{C2}-x_{C1}}(x - x_{C1}), & x \in [x_{C1}; x_{C2}] \\ d, & x \in [x_{C2}; x_B] \end{cases}$$

Следовательно эпюра  $\sigma_x$  2 раза меняет свой характер при движении от точки А к точке В.

### 9.3 Касательные напряжения $\tau_{xz}$

Касательные напряжения также определяются граничными условиями:

$$\tau_{xz,A} = n(p_A - \sigma_{z,A})$$

$$\tau_{xz,B} = -m(p_B - \sigma_{z,B})$$

Для уравновешивания силы  $Q$  и соблюдения граничных условий, выражение, описывающее  $\tau_{xz}$  можно разбить на две составляющие:

$$\tau_{xz} = \tilde{\tau}_{xz} + \Delta\tau_{xz}$$

Составляющая  $\tilde{\tau}_{xz}$  определяется аналогично  $\sigma_x$ :

$$\tilde{\tau}_{xz} = \frac{\tau_{xz,A}(x_B - x)D + \tau_{xz,B}(x - x_A)d}{B[d(x)]}$$

$$x \in [x_A, x_B]$$

Составляющая  $\Delta\tau_{xz}$  уравновешивает сдвигающую силу  $\Delta Q$ , которая определяется выражением:

$$\Delta Q = Q - (\tau_{xz,A}D + \tau_{xz,B}d)B/2$$

Выражение для определения  $\Delta\tau_{xz}$ :

$$\Delta\tau_{xz} = \frac{\Delta Q}{J_y} \Delta\tau(x)$$

Множитель  $\Delta\tau(x)$  зависит от текущей координаты  $x$ :

$$\Delta\tau(x) = \begin{cases} 0.5(x_A^2 - x^2), & x \in [x_A; x_{C1}] \\ \frac{3D(x_{C2} - x_{C1})(x_A^2 - x^2) + (D-d)(2x + x_{C1})(x - x_{C1})^2}{6[D(x_{C2} - x_{C1}) - (D-d)(x - x_{C1})]}, & x \in [x_{C1}; x_{C2}] \\ \frac{3D(x_A^2 - x^2) + (D-d)(3x^2 - x_{C1}^2 - x_{C1}x_{C2} - x_{C2}^2)}{6d}, & x \in [x_{C2}; 0] \\ 0.5(x_B^2 - x^2), & x \in [0; x_B] \end{cases}$$

Для устойчивости плотины также важно, чтобы полученные касательные напряжения  $\tau_{xz}$  не превышали предельных значений в каждой точке:

$$\tau_{xz,lim} = f\left(\frac{\sigma_{z,A}x_B - \sigma_{z,B}x_A}{B} + \frac{\sigma_{z,B} - \sigma_{z,A}}{B}x\right) + c$$

$$x \in [x_A, x_B]$$

В случае превышения можно увеличить ширину плотины по основанию В или ширину контрфорса d.

#### 9.4 Главные нормальные напряжения $\sigma_{1,2}$

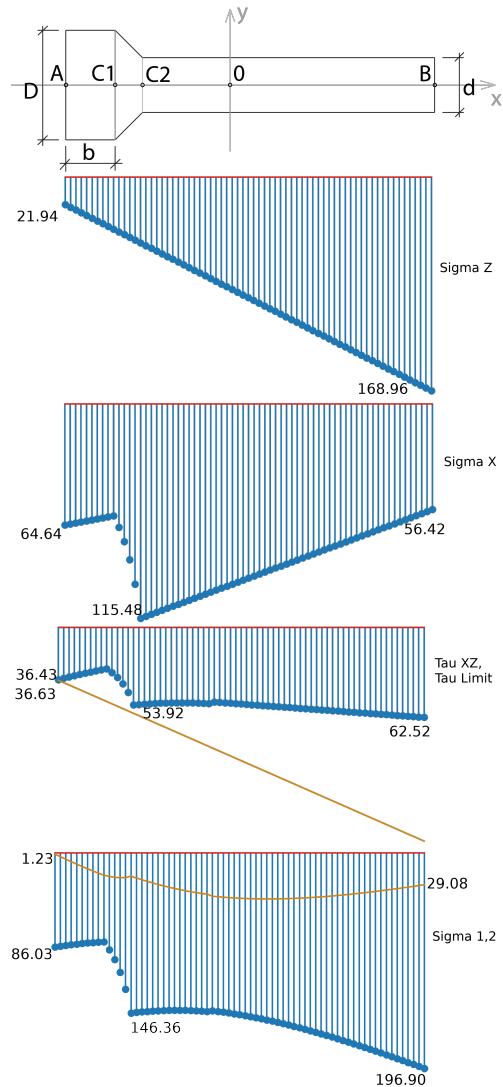
Главные нормальные напряжения  $\sigma_{1,2}$  связаны с напряжениями  $\sigma_z$ ,  $\sigma_x$ ,  $\tau_{xz}$  зависимостью:

$$\sigma_{1,2} = \frac{\sigma_x + \sigma_z \pm \sqrt{(\sigma_x - \sigma_z)^2 + 4\tau_{xz}^2}}{2}$$

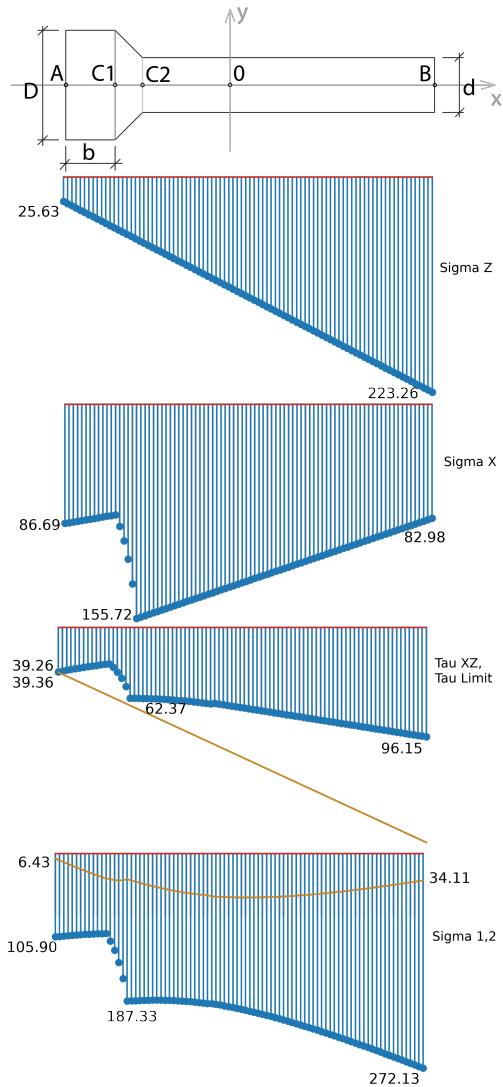
Из этого выражения также строится эпюра главных нормальных напряжений  $\sigma_{1,2}$ .

## 10 Эпюры напряжений в секции плотины

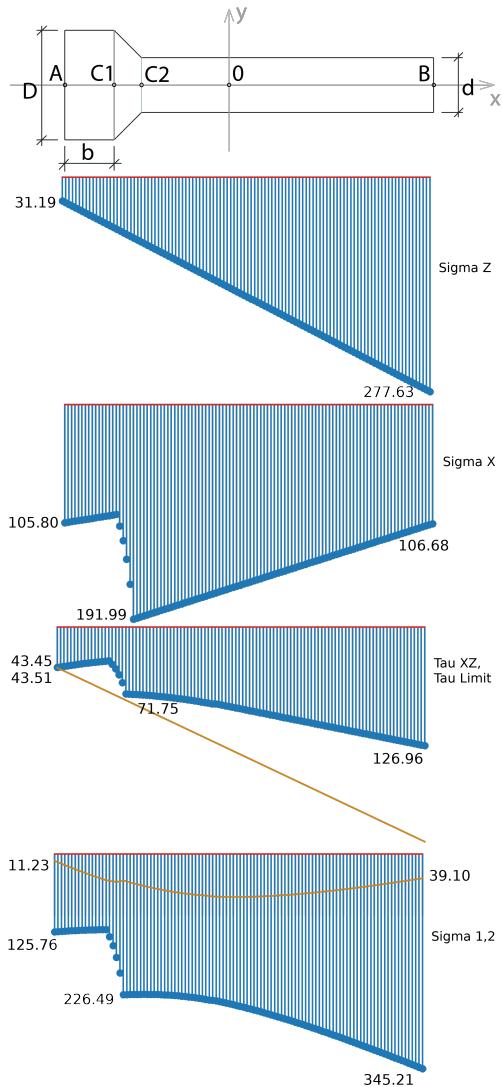
### 10.1 Для высоты плотины 60 метров



## 10.2 Для высоты плотины 80 метров



### 10.3 Для высоты плотины 100 метров



## 10.4 Рекомендации по назначению марки бетона для плотины

Максимальные сжимающие напряжения определяют марку бетона по прочности на сжатие. Каждому классу согласно СНиП соответствует определенные значения расчетного напряжения по сжатию.

Таблица допускаемых сжимающих напряжений в МПа:

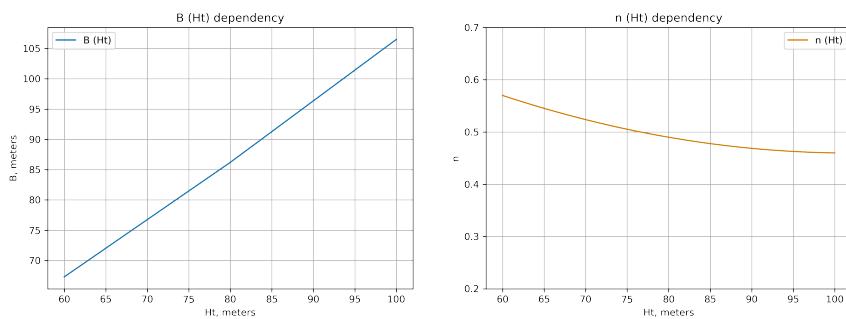
Класс бетона	B10	B12.5	B15	B20	B25	B30
$R_b$	6.0	7.5	8.5	11.5	14.5	17.0

Чтобы создать запас устойчивости можно сделать уклон верховой грани больше.

# Часть VII

## Рекомендации по назначению параметров контрфорсной плотины для различных её высот

Цель - связать условия отсутствия растягивающих напряжений, устойчивости и экономичности и посмотреть, какое будет оптимальное значение этих величин. Полученные графики могут упростить поиск решений для разных высот плотины.



Приложение к дипломному проекту

## Исходный код программы

Конструирование и статический расчет контрфорсной плотины с  
массивными оголовками

2020 г.

```

[1]: from math import sqrt
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import make_interp_spline, BSpline

float_formatter = "{:.2f}".format
np.set_printoptions(formatter={'float_kind':float_formatter})

import warnings
warnings.filterwarnings('ignore')

#SAVE_PLOTS = True
PLOTS_TRANSPARENT = True

[2]: def static_calc(B=60, n=0.4, Ht=60, d=10, D=20, sigma_z_allow=10,
    output=None, print_init=False, print_var=False, print_results=False, ↴
    plots=False, save_fig=None,
    plots_transparent=PLOTS_TRANSPARENT):

    const_pi = 3.1415
    b = 0.15 * Ht
    H2 = 105.2 - 98 # = 7.2 meters
    Bc = 10 #meters

    hw = 0.8 #meters

    Lw = 11 * hw #in meters
    dhw = const_pi * hw ** 2 / Lw #in meters
    m = B / Ht - n
    dc = hw + dhw + 0.6 #meters
    ec = 1 #meters

    W_ice = 10 #in tons / m
    h_ice = 1 #in meters
    H_allu = 24 #in meters - Alluvium layer thickness

    #density
    dens_c = 2.4 #tons / m ** 3 - Concrete
    dens_w = 1 #ton / m ** 3 - Water
    dens_allu = 1.05 #tons / m ** 3 - Alluvium

    sqrt1n = sqrt(1 + n ** 2)
    a = (D - d) / 2
    F = B * d + a * (2 * b + a * sqrt1n)

    x_cm = (0.5 * B ** 2 * d + a * b ** 2 + a ** 2 * (b + a * sqrt1n / 3) * ↴
    sqrt1n) / F

```

```

x_B = B - x_cm

Jy_list = [0, 0, 0, 0, 0, 0]
Jy_list[0] = B ** 3 * d / 12
Jy_list[1] = a * b ** 3 / 6
Jy_list[2] = a ** 4 / 18 * sqrtin ** 1.5
Jy_list[3] = (x_cm - 0.5 * B) ** 2 * B * d
Jy_list[4] = 2 * (x_cm - 0.5 * b) ** 2 * a * b
Jy_list[5] = a ** 2 * (x_cm - b - a * sqrtin / 3) ** 2 * sqrtin

Jy = 0
for item in Jy_list:
    Jy += item

W_A = Jy / x_cm
W_B = Jy / x_B

if print_init == True:
    print(f"Ht = {Ht}")
    print(f"B = {B}")
    print(f"n = {n}")
    print(f"d = {d}")
    print(f"D = {D}")
    print(f"b = {b}")
    print(f"m = {m}")
    print(f"dc = {dc}")
    print(f"F = {F}")
    print(f"x_A = {x_cm}")
    print(f"x_B = {x_B}")
    print(f"Jy = {Jy}")
    print(f"W_A = {W_A}")
    print(f"W_B = {W_B}", end="\n\n")

#Own weight

G1 = 0.5 * dens_c * B * Ht * d
M1 = dens_c * Ht * B * d * (B - 3 * x_cm + n * Ht) / 6

G2 = dens_c * Ht * (a * sqrtin + 2 * b) * a
M2_1 = b * (2 * x_cm - b - n * Ht)
M2_2 = a * sqrtin * (x_cm - b - (n * Ht) / 2 - (a * sqrtin) / 3)
M2 = -dens_c * Ht * a * (M2_1 + M2_2)

G3 = dens_c * Bc * D * dc
M3 = -G3 * (x_cm - n * Ht - ec)

a1 = (0.5 * Bc - ec)

```

```

G4 = 0.5 * dens_c * D * a1 ** 2 / n
M4 = -G4 * (x_cm - n * Ht + (2 * a1) / 3)

a2 = (0.5 * Bc + ec)
G5 = -0.5 * dens_c * n * D * a2 ** 2 / m ** 2
M5_1 = 2 * n * a2 / (3 * m)
M5 = -G5 * (x_cm - b - n * Ht + M5_1)

a_star = a2 * (b + a * sqrtin) * sqrtin / (m * b)
b_star = x_cm - b - n * Ht - a * sqrtin
n_1 = 1 - n * m / (n + m)
G6_1 = a_star - (2 * a * sqrtin * Ht) / (3 * B)
G6 = -dens_c * a ** 2 * G6_1

sqrtin_1 = sqrt(1 + n_1 ** 2)
M6_1 = n * (2 + n ** 2) / sqrtin + n_1 * (2 - n ** 2) / sqrtin_1
M6_2 = n * n_1 * a_star + (n + n_1) * b_star + 3 * a * M6_1 / 8
M6_2_2 = b_star + n * a / 2 + 2 * a / (3 * sqrtin)
M6_3 = a_star * M6_2_2 - 2 * a * M6_2 / 3
M6 = dens_c * a ** 2 * M6_3

if print_var == True:
    print("Own weight")
    print(f"G1 = {G1}")
    print(f"G2 = {G2}")
    print(f"G3 = {G3}")
    print(f"G4 = {G4}")
    print(f"G5 = {G5}")
    print(f"G6 = {G6}", end="\n\n")

    print(f"l1 = {M1 / G1}")
    print(f"l2 = {M2 / G2}")
    print(f"l3 = {M3 / G3}")
    print(f"l4 = {M4 / G4}")
    print(f"l5 = {M5 / G5}")
    print(f"l6 = {M6 / G6}", end="\n\n")

    print(f"a1 = {a1}")
    print(f"a2 = {a2}")
    print(f"a* = {a_star}")
    print(f"b* = {b_star}")
    print(f"n1 = {n_1}", end="\n\n")

    print(f"M1 = {M1}")
    print(f"M2 = {M2} = {M2_1} and {M2_2}")
    print(f"M3 = {M3}")
    print(f"M4 = {M4}")

```

```

    print(f"M5 = {M5}, M5_1 = {M5_1}")
    print(f"M6 = {M6} brackets = {M6_1}, {M6_2}, {M6_2_2}, {M6_3}", end="\n\n")
→end="\\n\\n")

#Water pressure
#upper horizontal

H1 = Ht
W_uphor = 0.5 * dens_w * H1 ** 2 * D
M_uphor = dens_w * H1 ** 3 * D / 6

#upper vertical
W_upvert = n * W_uphor
M_upvert = -dens_w * n * D * H1 ** 2 * (3 * x_cm - n * H1) / 6

#lower horizontal
W_lowhor = -0.5 * dens_w * H2 ** 2 * D
M_lowhor = W_lowhor * H2 / 3

#lower vertical
W_lowvert = (n - B * d / (Ht * D)) * W_lowhor
M_lowvert_1 = x_cm - b - a / 2 - n * H2 / 3
M_lowvert_2 = x_B - m * H2
M_lowvert_3 = 2 * n * a * M_lowvert_1 + m * M_lowvert_2 * d
M_lowvert = M_lowvert_3 * W_lowvert / D

#Anti-pressure
U1 = -dens_w * H2 * (D * b + a * (D + d) / 2 + (B - b - a) * d)
M_U1 = 0
U2 = -dens_w * (H1 - H2) * (b + a) * D
M_U2 = -U2 * (x_cm - (b + a) / 2)

#Alluvium pressure
W_alluhor = 0.5 * dens_allu * H_allu ** 2 * D
M_alluhor = W_alluhor * H_allu / 3

W_alluvert = 0.5 * dens_allu * n * H_allu ** 2 * D
M_alluvert = -W_alluvert * (x_cm - n * H_allu / 3)

#Wave pressure
W_wavhor = dens_w * hw * (Lw / const_pi + hw / 2) * D / 2
M_wavhor = W_wavhor * (H1 - Lw / (2 * const_pi) + 3 * hw / 8)

W_wavvert = W_wavhor * n

if print_var == True:
    print("Water pressure")

```

```

print(f"W upper horizontal = {W_uphor}")
print(f"W upper vertical = {W_upvert}")
print(f"W lower horizontal = {W_lowhor}")
print(f"W lower vertical = {W_lowvert}")

print(f"M upper horizontal = {M_uphor}")
print(f"M upper vertical = {M_upvert}")
print(f"M lower horizontal = {M_lowhor}")
print(f"M lower vertical = {M_lowvert}")

print(f"Suspending force = {U1}")
print(f"Filtering part = {U2}")
print(f"Filtering part moment = {M_U2}")

print(f"Alluvium vertical = {W_alluvert}")
print(f"Alluvium vertical moment = {M_alluvert}")
print(f"Alluvium horizontal = {W_alluhor}")
print(f"Alluvium horizontal moment = {M_alluvert}")

print(f"Wave pressure horizontal part = {W_wavhor}")
print(f"Wave pressure moment = {M_wavhor}")
print(f"Wave pressure vertical part = {W_wavvert}")

print("")

#Construction period - reservoir empty

N = G1 + G2 + G3 + G4 + G5 + G6
Qsummer = None
Msummer = M1 + M2 + M3 + M4 + M5 + M6

if print_results == True:
    print("Summary")
    print("Construction period")
    print(f"N = {G1 + G2 + G3 + G4 + G5 + G6}")
    print(f"M = {M1 + M2 + M3 + M4 + M5 + M6}")

#Working period - reservoir full

M_water = M_uphor + M_upvert + M_lowhor + M_lowvert
N = G1 + G2 + G3 + G4 + G5 + G6 + W_upvert + W_lowvert + U1 + U2 + W_wavvert
→+ W_alluvert
Qsummer = W_uphor + W_lowhor + W_alluhor + W_wavhor
Msummer = M1 + M2 + M3 + M4 + M5 + M6 + M_water + M_U2 + M_alluhor + 
→M_alluvert + M_wavhor

```

```

if print_results == True:
    print("Working period")
    print(f"N = {N}")
    print(f"Q = {Qsummer}")
    print(f"M = {Msummer}", end="\n\n")

#criteria
k3 = 1.25
f = 0.75
c = 20
no_stretch_eq = sigma_z_allow * B ** 2 - N * B + 6 * Msummer
sustainability = f * N + c * B - k3 * Qsummer
if print_results == True:
    print(f"Stretch criteria = {no_stretch_eq}")
    print(f"Sustainability criteria = {sustainability}", end="\n\n")

#Stresses

#sigma Z
coord = np.arange(-x_cm, -x_cm + B + 1, 1)
sigma_z = N / F + Msummer * coord / Jy
sigma_zA = sigma_z[0]
sigma_zB = sigma_z[-1]

#sigma X
p_A = dens_w * Ht + dens_allu * H_allu
p_B = dens_w * H2
sigma_xA = (1 - n ** 2) * p_A + n ** 2 * sigma_zA
sigma_xB = (1 - m ** 2) * p_B + m ** 2 * sigma_zB

C1 = int(b)
C2 = int(b + a)
C0 = int(x_cm)
d_x_sigma_x = np.zeros(len(coord))

d_x_sigma_x[0:C1] = D
d_x_sigma_x[C1:C2] = D - (D - d) * (coord[C1:C2] - coord[C1]) / (coord[C2] -
→coord[C1])
d_x_sigma_x[C2:] = d

sigma_x = (sigma_xA * (coord[-1] - coord) * D + sigma_xB * (coord -
→coord[0]) * d) / (B * d_x_sigma_x)

#tau XZ
tau_xzA = n * (p_A - sigma_zA)

```

```

tau_xzB = -m * (p_B - sigma_zB)
d_x_tau_wave = d_x_sigma_x
deltaTau = np.zeros(len(coord))
tau_wave = np.zeros(len(coord))
tau_wave = (tau_xzA * (coord[-1] - coord) * D + tau_xzB * (coord - coord[0]) *
→* d) / (B * d_x_tau_wave)

deltaQ = Qsummer - (tau_xzA * D + tau_xzB * d) * B / 2

deltaTau[0:C1] = 0.5 * (coord[0] ** 2 - coord[0:C1] ** 2)
deltaTau[C1:C2] = 3 * D * (coord[C2] - coord[C1]) * (coord[0] ** 2 -
→coord[C1:C2] ** 2)
deltaTau[C1:C2] += (D - d) * (2 * coord[C1:C2] + coord[C1]) * (coord[C1:C2] -
→coord[C1]) ** 2
deltaTau[C1:C2] /= 6 * (D * (coord[C2] - coord[C1]) - (D - d) * (coord[C1:
→C2] - coord[C1]))
deltaTau[C2:C0] = (D - d) * (3 * coord[C2:C0] ** 2 - coord[C1] ** 2 -
→coord[C1] * coord[C2] - coord[C2] ** 2) / (6 * d)
deltaTau[C2:C0] += 3 * D * (coord[0] ** 2 - coord[C2:C0] ** 2) / (6 * d)
deltaTau[C0:] = 0.5 * (coord[-1] ** 2 - coord[C0:])
deltaTau *= deltaQ / Jy

tau_xz = deltaTau + tau_wave

tau_xzlim = (sigma_zA * coord[-1] - sigma_zB * coord[0]) / B + (sigma_zB -
→sigma_zA) * coord / B
tau_xzlim = f * tau_xzlim + c
tau_xz_check = len(tau_xz) - np.sum(tau_xzlim > tau_xz)

sigma_1 = (sigma_x + sigma_z + np.sqrt((sigma_x - sigma_z) ** 2 + 4 * tau_xz *
→** 2)) / 2
sigma_2 = (sigma_x + sigma_z - np.sqrt((sigma_x - sigma_z) ** 2 + 4 * tau_xz *
→** 2)) / 2
tau_max = 0.5 * (sigma_1 - sigma_2)

if print_results == True:
    print("Stresses:")
    print(f"sigma z,A = {sigma_zA}")
    print(f"sigma z,B = {sigma_zB}")
    print(f"coordinates = {coord}")
    print(f"sigma z = {sigma_z}", end="\n\n")
    print(f"sigma x,A = {sigma_xA}")
    print(f"sigma x,B = {sigma_xB}")
    print(f"sigma x = {sigma_x}", end="\n\n")

```

```

print(f"p_a and p_b = {p_A, p_B}")
print(f"tau_xz,A = {tau_xzA}")
print(f"tau_xz,B = {tau_xzB}")
print(f"delta_Q = {deltaQ}")
print(f"tau_xz = {tau_xz}")
print(f"tau_xz_limit = {tau_xzlim}")
print(f"tau_check = {tau_xz_check}, if zero - everything OK", end="\n\n")
print(f"sigma_1 = {sigma_1}")
print(f"sigma_2 = {sigma_2}")
print(f"tau_max = {tau_max}", end="\n\n")

if plots == True:
    fig, ((ax_sigma_z, ax_sigma_x), (ax_tau_xz, ax_sigma_12)) = plt.
    →subplots(figsize=(15,9), nrows=2, ncols=2)
    color1 = "#399aef"
    color2 = "#d3840c"

    ax_sigma_z.stem(coord, sigma_z, use_line_collection=True, label="Sigma_"
    →Z")
    ax_sigma_z.legend()
    ax_sigma_z.set_title("Sigma Z")

    ax_sigma_x.stem(coord, sigma_x, use_line_collection=True, label="Sigma_"
    →X")
    ax_sigma_x.legend()
    ax_sigma_x.set_title("Sigma X")

    ax_tau_xz.stem(coord, tau_xz, use_line_collection=True, label="Tau_XZ")
    ax_tau_xz.plot(coord, tau_xzlim, color=color2, label="Tau_XZ_Limit")
    ax_tau_xz.legend()
    ax_tau_xz.set_title("Tau_XZ")

    ax_sigma_12.stem(coord, sigma_1, use_line_collection=True, label="Sigma_"
    →1")
    ax_sigma_12.plot(coord, sigma_2, color=color2, label="Sigma_2")
    ax_sigma_12.legend()
    ax_sigma_12.set_title("Sigma_1 & Sigma_2")

    if save_fig != None:
        if _plots_transparent == True:
            save_fig = "t_" + save_fig
        plt.savefig(save_fig + ".png", dpi=300, transparent=_plots_transparent)

    plt.show()

#return values

```

```

if output == None:
    return None
if output == "G":
    return G1 + G2 + G3 + G4 + G5 + G6
if output == "no_stretch_eq":
    return no_stretch_eq
if output == "sustainability":
    return sustainability
if output == "sigma_zA":
    return sigma_zA
if output == "sigma_xc1":
    return sigma_x[C1]
if output == "tau_xz_check":
    return tau_xz_check
if output == "fullset":
    return no_stretch_eq, sustainability, sigma_zA

static_calc(output="fullset", print_init=True, print_var=True, □
→print_results=True, plots=True)

```

```

Ht = 60
B = 60
n = 0.4
d = 10
D = 20
b = 9.0
m = 0.6
dc = 1.6284727272727273
F = 716.9258240356726
x_A = 26.077545434775875
x_B = 33.92245456522413
Jy = 238069.49037330007
W_A = 9129.29059863974
W_B = 7018.0502391285945

Own weight
G1 = 43200.0
G2 = 16837.318661136844
G3 = 781.6669090909091
G4 = 960.0
G5 = -959.999999999999
G6 = -817.4798512374274

l1 = 1.9224545652241243
l2 = -8.127912362768209
l3 = -1.0775454347758746

```

```

14 = -4.744212101442541
15 = 4.2557878985574575
16 = 9.042923721469545

a1 = 4.0
a2 = 6.0
a* = 17.21477405871346
b* = -12.30761937235863
n1 = 0.76

M1 = 83050.03721768217
M2 = -136852.250501722 = 172.39581782596574 and 17.67675231531484
M3 = -842.2816094062778
M4 = -4554.443617384839
M5 = -4085.556382615159, M5_1 = 2.6666666666666674
M6 = -7392.407938578328 brackets = 1.9155570731866105, -5.451877645862224,
-8.212697069407767, -123.20679897630546

Water pressure
W upper horizontal = 36000.0
W upper vertical = 14400.0
W lower horizontal = -518.4000000000004
W lower vertical = 51.84000000000003
M upper horizontal = 720000.0
M upper vertical = -260316.65426077263
M lower horizontal = -1244.1600000000014
M lower vertical = 601.5640844661223
Suspending force = -5148.000000000002
Filtering part = -14783.999999999998
Filtering part moment = 282042.4317077265
Alluvium vertical = 2419.200000000003
Alluvium vertical moment = -55345.35791580981
Alluvium horizontal = 6048.000000000001
Alluvium horizontal moment = -55345.35791580981
Wave pressure horizontal part = 25.609676905936656
Wave pressure moment = 1508.3944808575136
Wave pressure vertical part = 10.243870762374662

Summary
Construction period
N = 60001.50571899033
M = -70676.90283202444
Working period
N = 56950.7895897527
Q = 41555.209676905935
M = 664953.3152644433

Stretch criteria = 608672.5162014975

```

Sustainability criteria = -8030.9199038178995

Stresses:

sigma z,A = 6.6001443650442155  
sigma z,B = 174.18650267289206  
coordinates = [-26.08 -25.08 -24.08 -23.08 -22.08 -21.08 -20.08 -19.08 -18.08  
-17.08  
-16.08 -15.08 -14.08 -13.08 -12.08 -11.08 -10.08 -9.08 -8.08 -7.08 -6.08  
-5.08 -4.08 -3.08 -2.08 -1.08 -0.08 0.92 1.92 2.92 3.92 4.92 5.92 6.92  
7.92 8.92 9.92 10.92 11.92 12.92 13.92 14.92 15.92 16.92 17.92 18.92  
19.92 20.92 21.92 22.92 23.92 24.92 25.92 26.92 27.92 28.92 29.92 30.92  
31.92 32.92 33.92]  
sigma z = [6.60 9.39 12.19 14.98 17.77 20.57 23.36 26.15 28.94 31.74 34.53 37.32  
40.12 42.91 45.70 48.50 51.29 54.08 56.88 59.67 62.46 65.26 68.05 70.84  
73.63 76.43 79.22 82.01 84.81 87.60 90.39 93.19 95.98 98.77 101.57 104.36  
107.15 109.95 112.74 115.53 118.32 121.12 123.91 126.70 129.50 132.29  
135.08 137.88 140.67 143.46 146.26 149.05 151.84 154.63 157.43 160.22  
163.01 165.81 168.60 171.39 174.19]

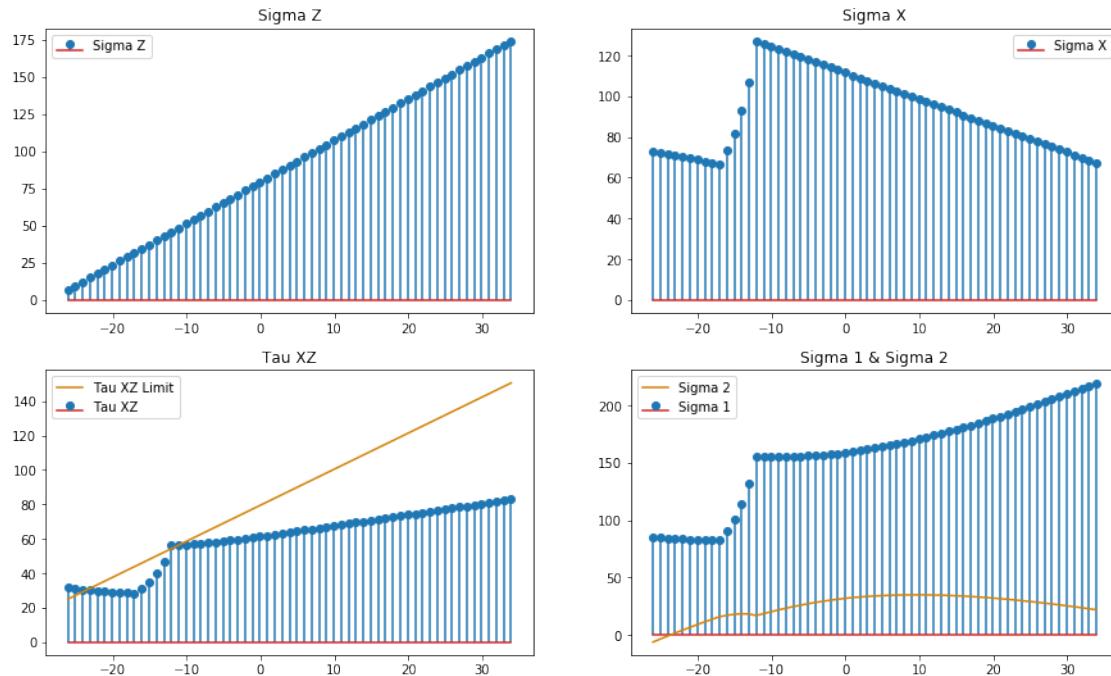
sigma x,A = 72.62402309840708  
sigma x,B = 67.31514096224114  
sigma x = [72.62 71.97 71.33 70.68 70.03 69.38 68.73 68.08 67.43 66.78 73.48  
81.85  
92.62 106.97 127.06 125.76 124.47 123.17 121.87 120.57 119.27 117.97  
116.67 115.37 114.07 112.78 111.48 110.18 108.88 107.58 106.28 104.98  
103.68 102.38 101.09 99.79 98.49 97.19 95.89 94.59 93.29 91.99 90.70  
89.40 88.10 86.80 85.50 84.20 82.90 81.60 80.30 79.01 77.71 76.41 75.11  
73.81 72.51 71.21 69.91 68.61 67.32]

p\_a and p\_b = (85.2, 7.2000000000000003)  
tau xz,A = 31.439942253982316  
tau xz,B = 100.19190160373525  
delta Q = -7366.326156604031  
tau xz = [31.44 30.96 30.51 30.09 29.70 29.35 29.02 28.73 28.46 28.23 31.17  
34.94  
39.89 46.61 56.12 56.38 56.68 57.00 57.36 57.75 58.17 58.62 59.10 59.61  
60.15 60.72 61.24 61.88 62.52 63.16 63.79 64.43 65.07 65.71 66.34 66.98  
67.62 68.26 68.89 69.53 70.17 70.80 71.44 72.08 72.72 73.35 73.99 74.63  
75.27 75.90 76.54 77.18 77.82 78.45 79.09 79.73 80.36 81.00 81.64 82.28  
82.91]  
tau xz limit = [24.95 27.04 29.14 31.23 33.33 35.42 37.52 39.61 41.71 43.80  
45.90 47.99  
50.09 52.18 54.28 56.37 58.47 60.56 62.66 64.75 66.85 68.94 71.04 73.13  
75.23 77.32 79.42 81.51 83.61 85.70 87.79 89.89 91.98 94.08 96.17 98.27  
100.36 102.46 104.55 106.65 108.74 110.84 112.93 115.03 117.12 119.22  
121.31 123.41 125.50 127.60 129.69 131.79 133.88 135.98 138.07 140.17  
142.26 144.36 146.45 148.55 150.64]  
tau check = 5, if zero - everything OK

```

sigma 1 = [85.20 84.70 84.24 83.83 83.46 83.14 82.88 82.68 82.54 82.48 90.76
101.02
114.12 131.49 155.70 155.48 155.34 155.28 155.30 155.40 155.60 155.88
156.26 156.74 157.31 157.99 158.68 159.56 160.51 161.53 162.62 163.78
165.01 166.31 167.67 169.09 170.58 172.12 173.72 175.38 177.08 178.84
180.65 182.50 184.40 186.34 188.33 190.35 192.40 194.50 196.62 198.78
200.97 203.18 205.43 207.70 209.99 212.31 214.65 217.01 219.39]
sigma 2 = [-5.98 -3.33 -0.73 1.83 4.34 6.80 9.21 11.55 13.83 16.03 17.25 18.16
18.61
18.39 17.07 18.78 20.42 21.97 23.45 24.83 26.13 27.34 28.46 29.48 30.40
31.22 32.02 32.63 33.18 33.65 34.05 34.38 34.65 34.85 34.98 35.05 35.06
35.01 34.91 34.75 34.53 34.27 33.96 33.60 33.19 32.74 32.26 31.73 31.17
30.57 29.94 29.27 28.58 27.86 27.11 26.33 25.53 24.71 23.86 23.00 22.11]
tau max = [45.59 44.02 42.49 41.00 39.56 38.17 36.83 35.56 34.36 33.22 36.75
41.43
47.76 56.55 69.31 68.35 67.46 66.65 65.93 65.28 64.73 64.27 63.90 63.63
63.46 63.38 63.33 63.46 63.67 63.94 64.29 64.70 65.18 65.73 66.34 67.02
67.76 68.55 69.41 70.31 71.27 72.29 73.35 74.45 75.61 76.80 78.03 79.31
80.62 81.96 83.34 84.75 86.19 87.66 89.16 90.68 92.23 93.80 95.39 97.01
98.64]

```



[2]: (608672.5162014975, -8030.9199038178995, 6.6001443650442155)

0.1 Ht = 60 m

```
[3]: def plot_bn(Ht_plot=60, B_range=(40, 91), print_init=False, save_fig=None, ↴_plots_transparent=PLOTS_TRANSPARENT):
    B_array = np.arange(B_range[0], B_range[1], 1)
    n_array = np.linspace(0.45, 0.7, B_array.size)
    B_graph_array = np.zeros((3, n_array.size))
    sigma_z_allow = 30

    super_array = np.zeros((n_array.size, 4, B_array.size))

    if print_init == True:
        print(n_array)
        print(B_array)
        print(sigma_z_allow)

    for i in super_array:
        i[0] = B_array

#sustain
for j, n in enumerate(n_array):

    for i, B in enumerate(B_array):
        super_array[j][1][i] = static_calc(Ht=Ht_plot, n=n, B=B, ↴sigma_z_allow=20, output="sustainability")
        super_array[j][2][i] = static_calc(Ht=Ht_plot, n=n, B=B, ↴sigma_z_allow=20, output="no_stretch_eq")
        super_array[j][3][i] = static_calc(Ht=Ht_plot, n=n, B=B, ↴sigma_z_allow=20, output="sigma_zA")
        if super_array[j][1][i] > 0 and super_array[j][1][i-1] < 0:
            #print(str(n)[:4], super_array[j][0][i-1], super_array[j][0][i], ↴super_array[j][1][i-1], super_array[j][1][i])
            B_graph_array[0][j] = super_array[j][0][i-1] - ↴super_array[j][1][i-1] / (super_array[j][1][i] - super_array[j][1][i-1])
            B_graph_array[1][j] = super_array[j][0][i-1] - ↴super_array[j][2][i-1] / (super_array[j][2][i] - super_array[j][2][i-1])
            B_graph_array[2][j] = super_array[j][0][i-1] - ↴super_array[j][3][i-1] / (super_array[j][3][i] - super_array[j][3][i-1])

#print(B_graph_array)

fig, ax = plt.subplots()
ax.plot(n_array, B_graph_array[0], label="Sustainability")
ax.plot(n_array, B_graph_array[1], label="No stretch stress")
ax.grid()
```

```

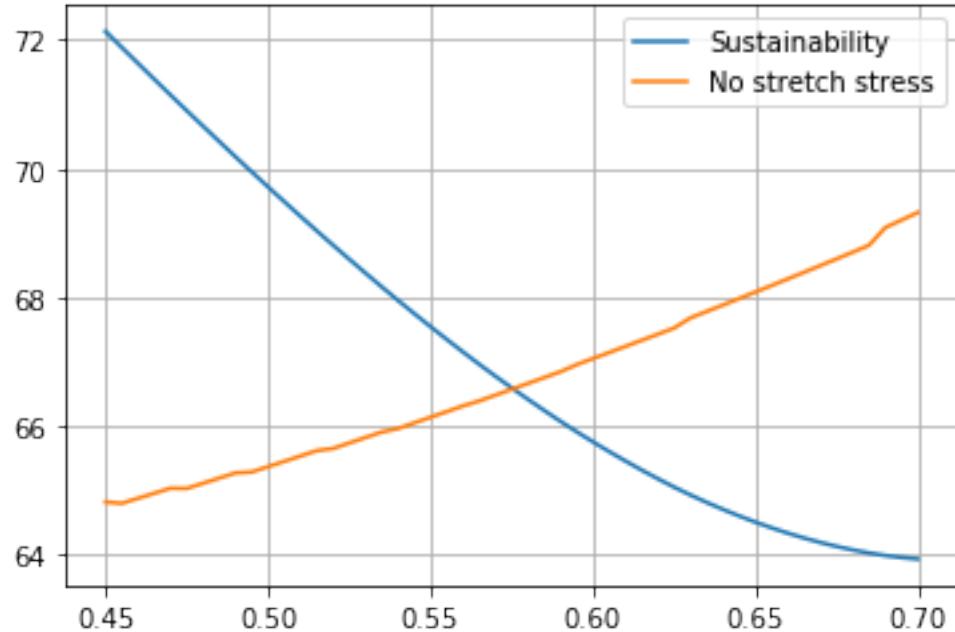
ax.legend()

if save_fig != None:
    if _plots_transparent == True:
        save_fig = "t_" + save_fig
    plt.savefig(save_fig + ".png", dpi=300, transparent=_plots_transparent)

plt.show()

plot_bn(Ht_plot=60, save_fig="Bn60")

```



0.1.1 B = 67.3 m, n=0.57

[4]: static\_calc(n=0.57, B=67.3, sigma\_z\_allow=20, output="tau\_xz\_check", plots=True,  
→print\_init=True, print\_results=True, save\_fig="sigmatau60")

```

Ht = 60
B = 67.3
n = 0.57
d = 10
D = 20
b = 9.0
m = 0.5516666666666666
dc = 1.6284727272727273
F = 791.7760751319564

```

```
x_A = 29.510412825949793
x_B = 37.789587174050205
Jy = 332444.39519283274
W_A = 11265.325129592897
W_B = 8797.248661692698
```

Summary

Construction period

N = 64370.67297101742

M = 75023.93723390526

Working period

N = 67890.27848685381

Q = 41555.209676905935

M = 718760.5226923459

Stretch criteria = -165866.8060111869

Sustainability criteria = 319.69676900794

Stresses:

sigma z,A = 21.941381608363642

sigma z,B = 168.96060121268093

coordinates = [-29.51 -28.51 -27.51 -26.51 -25.51 -24.51 -23.51 -22.51 -21.51  
-20.51

-19.51 -18.51 -17.51 -16.51 -15.51 -14.51 -13.51 -12.51 -11.51 -10.51

-9.51 -8.51 -7.51 -6.51 -5.51 -4.51 -3.51 -2.51 -1.51 -0.51 0.49 1.49

2.49 3.49 4.49 5.49 6.49 7.49 8.49 9.49 10.49 11.49 12.49 13.49 14.49

15.49 16.49 17.49 18.49 19.49 20.49 21.49 22.49 23.49 24.49 25.49 26.49

27.49 28.49 29.49 30.49 31.49 32.49 33.49 34.49 35.49 36.49 37.49 38.49]

sigma z = [21.94 24.10 26.27 28.43 30.59 32.75 34.91 37.08 39.24 41.40 43.56  
45.72

47.89 50.05 52.21 54.37 56.53 58.70 60.86 63.02 65.18 67.34 69.51 71.67

73.83 75.99 78.15 80.32 82.48 84.64 86.80 88.96 91.13 93.29 95.45 97.61

99.78 101.94 104.10 106.26 108.42 110.59 112.75 114.91 117.07 119.23

121.40 123.56 125.72 127.88 130.04 132.21 134.37 136.53 138.69 140.85

143.02 145.18 147.34 149.50 151.66 153.83 155.99 158.15 160.31 162.47

164.64 166.80 168.96]

sigma x,A = 64.64727488455735

sigma x,B = 56.429592304062595

sigma x = [65.32 64.78 64.24 63.70 63.15 62.61 62.07 61.53 60.99 60.45 66.56  
74.21

84.03 97.14 115.48 114.40 113.32 112.23 111.15 110.07 108.99 107.90

106.82 105.74 104.65 103.57 102.49 101.41 100.32 99.24 98.16 97.08 95.99

94.91 93.83 92.75 91.66 90.58 89.50 88.41 87.33 86.25 85.17 84.08 83.00

81.92 80.84 79.75 78.67 77.59 76.50 75.42 74.34 73.26 72.17 71.09 70.01

68.93 67.84 66.76 65.68 64.60 63.51 62.43 61.35 60.26 59.18 58.10 57.02]

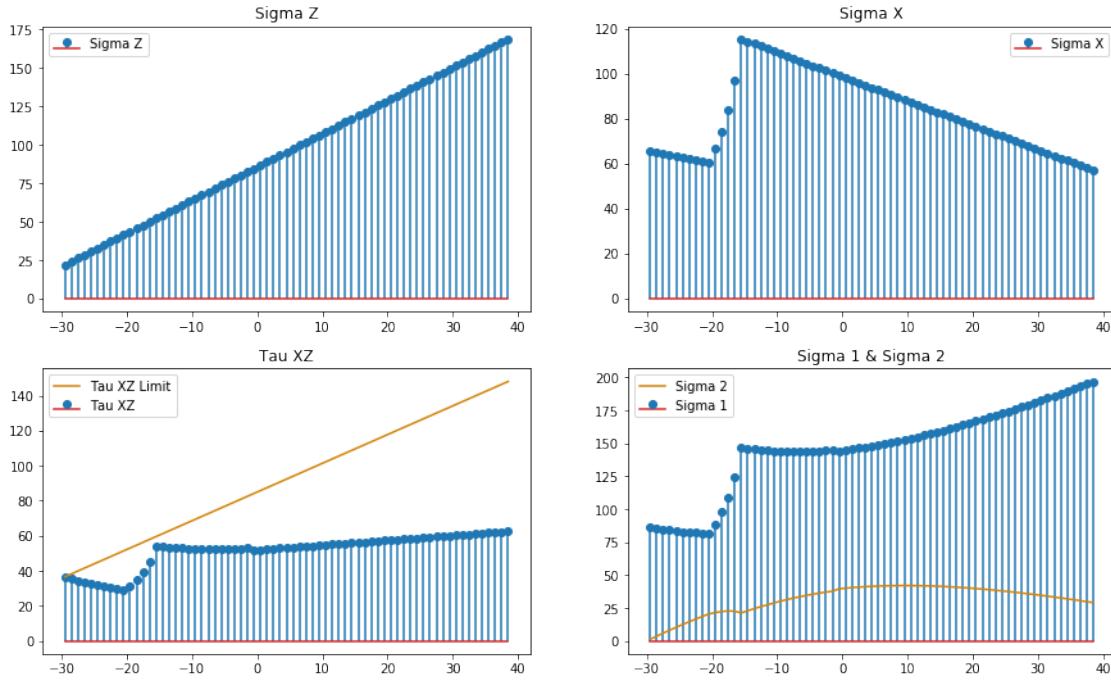
p\_a and p\_b = (85.2, 7.200000000000003)

```

tau xz,A = 36.057412483232724
tau xz,B = 89.23793166899563
delta Q = -12739.992930926717
tau xz = [36.43 35.45 34.50 33.59 32.72 31.89 31.10 30.35 29.63 28.95 31.50
34.82
39.24 45.28 53.92 53.60 53.32 53.07 52.87 52.70 52.57 52.48 52.43 52.41
52.43 52.50 52.60 52.74 52.91 51.85 52.12 52.39 52.67 52.94 53.22 53.49
53.76 54.04 54.31 54.58 54.86 55.13 55.40 55.68 55.95 56.22 56.50 56.77
57.05 57.32 57.59 57.87 58.14 58.41 58.69 58.96 59.23 59.51 59.78 60.06
60.33 60.60 60.88 61.15 61.42 61.70 61.97 62.24 62.52]
tau xz limit = [36.63 38.27 39.90 41.54 43.18 44.82 46.46 48.10 49.73 51.37
53.01 54.65
56.29 57.93 59.56 61.20 62.84 64.48 66.12 67.76 69.40 71.03 72.67 74.31
75.95 77.59 79.23 80.86 82.50 84.14 85.78 87.42 89.06 90.69 92.33 93.97
95.61 97.25 98.89 100.52 102.16 103.80 105.44 107.08 108.72 110.36 111.99
113.63 115.27 116.91 118.55 120.19 121.82 123.46 125.10 126.74 128.38
130.02 131.65 133.29 134.93 136.57 138.21 139.85 141.48 143.12 144.76
146.40 148.04]
tau check = 0, if zero - everything OK

sigma 1 = [86.03 85.31 84.63 84.00 83.42 82.90 82.43 82.02 81.68 81.40 88.60
97.59
109.16 124.63 146.36 145.81 145.33 144.90 144.55 144.25 144.03 143.88
143.81 143.81 143.90 144.06 144.31 144.64 145.06 144.30 144.91 145.57
146.28 147.05 147.86 148.72 149.63 150.59 151.60 152.65 153.74 154.87
156.05 157.27 158.52 159.82 161.14 162.51 163.90 165.33 166.78 168.27
169.78 171.32 172.89 174.48 176.09 177.73 179.38 181.06 182.75 184.46
186.20 187.94 189.71 191.48 193.28 195.08 196.90]
sigma 2 = [1.23 3.57 5.87 8.12 10.32 12.47 14.56 16.59 18.55 20.45 21.53 22.34
22.76
22.55 21.33 22.96 24.52 26.03 27.46 28.83 30.14 31.36 32.52 33.59 34.59
35.50 36.33 37.08 37.74 39.58 40.05 40.47 40.84 41.15 41.42 41.63 41.80
41.92 42.00 42.03 42.02 41.96 41.86 41.72 41.55 41.34 41.09 40.80 40.49
40.14 39.76 39.36 38.92 38.46 37.98 37.47 36.93 36.38 35.80 35.21 34.59
33.96 33.31 32.64 31.95 31.26 30.54 29.82 29.08]
tau max = [42.40 40.87 39.38 37.94 36.55 35.21 33.94 32.72 31.56 30.48 33.53
37.62
43.20 51.04 62.51 61.43 60.40 59.44 58.54 57.71 56.95 56.26 55.65 55.11
54.65 54.28 53.99 53.78 53.66 52.36 52.43 52.55 52.72 52.95 53.22 53.54
53.92 54.33 54.80 55.31 55.86 56.46 57.09 57.77 58.49 59.24 60.03 60.85
61.71 62.59 63.51 64.46 65.43 66.43 67.46 68.51 69.58 70.67 71.79 72.93
74.08 75.25 76.44 77.65 78.88 80.11 81.37 82.63 83.91]

```



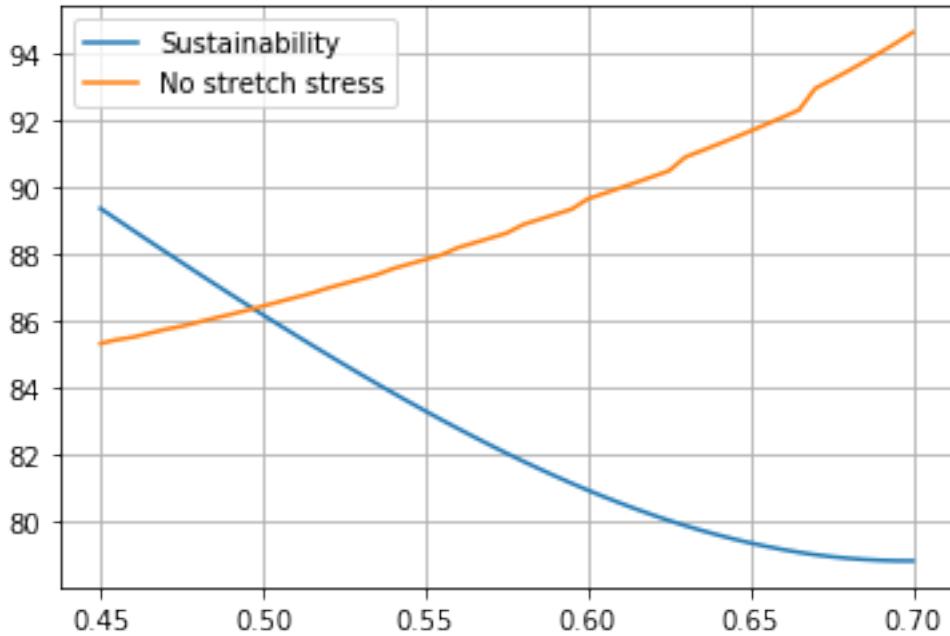
[4]: 0

0.2 Ht = 80 m

[5]: `plot_bn(Ht_plot=80, print_init=True, save_fig="Bn80")`

```
[0.45 0.46 0.46 0.47 0.47 0.47 0.48 0.48 0.49 0.49 0.50 0.51 0.51 0.52
 0.52 0.53 0.53 0.54 0.54 0.55 0.55 0.55 0.56 0.56 0.57 0.57 0.57 0.58 0.58
 0.59 0.59 0.60 0.60 0.61 0.61 0.62 0.62 0.63 0.64 0.64 0.65 0.65 0.66
 0.66 0.67 0.67 0.68 0.68 0.68 0.69 0.69 0.70]
[40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87
 88 89 90]
```

30



0.2.1 B=86.2, n=0.49

```
[6]: static_calc(B=86.4, n=0.49, Ht=80, output="tau_xz_check", print_init=True, plots=True, print_results=True, save_fig="sigmatau80")
```

```
Ht = 80
B = 86.4
n = 0.49
d = 10
D = 20
b = 12.0
m = 0.5900000000000001
dc = 1.62847272727273
F = 1011.8399443246569
x_A = 37.98085891032829
x_B = 48.41914108967172
Jy = 701428.9329324636
W_A = 18467.95867856799
W_B = 14486.604205420019
```

```
Summary
Construction period
N = 110889.86620996582
M = -10422.982080362695
Working period
```

N = 113235.05495164974  
Q = 69555.20967690594  
M = 1593344.6219838136

Stretch criteria = -148791.41591965593  
Sustainability criteria = -289.7208823951223

Stresses:

sigma z,A = 25.63388276082648  
sigma z,B = 223.26043565059553  
coordinates = [-37.98 -36.98 -35.98 -34.98 -33.98 -32.98 -31.98 -30.98 -29.98  
-28.98  
-27.98 -26.98 -25.98 -24.98 -23.98 -22.98 -21.98 -20.98 -19.98 -18.98  
-17.98 -16.98 -15.98 -14.98 -13.98 -12.98 -11.98 -10.98 -9.98 -8.98 -7.98  
-6.98 -5.98 -4.98 -3.98 -2.98 -1.98 -0.98 0.02 1.02 2.02 3.02 4.02 5.02  
6.02 7.02 8.02 9.02 10.02 11.02 12.02 13.02 14.02 15.02 16.02 17.02 18.02  
19.02 20.02 21.02 22.02 23.02 24.02 25.02 26.02 27.02 28.02 29.02 30.02  
31.02 32.02 33.02 34.02 35.02 36.02 37.02 38.02 39.02 40.02 41.02 42.02  
43.02 44.02 45.02 46.02 47.02 48.02 49.02]  
sigma z = [25.63 27.91 30.18 32.45 34.72 36.99 39.26 41.53 43.81 46.08 48.35  
50.62  
52.89 55.16 57.44 59.71 61.98 64.25 66.52 68.79 71.07 73.34 75.61 77.88  
80.15 82.42 84.69 86.97 89.24 91.51 93.78 96.05 98.32 100.60 102.87  
105.14 107.41 109.68 111.95 114.23 116.50 118.77 121.04 123.31 125.58  
127.85 130.13 132.40 134.67 136.94 139.21 141.48 143.76 146.03 148.30  
150.57 152.84 155.11 157.38 159.66 161.93 164.20 166.47 168.74 171.01  
173.29 175.56 177.83 180.10 182.37 184.64 186.92 189.19 191.46 193.73  
196.00 198.27 200.54 202.82 205.09 207.36 209.63 211.90 214.17 216.45  
218.72 220.99 223.26]

sigma x,A = 86.09617525087444  
sigma x,B = 82.41063764997232  
sigma x = [86.69 86.17 85.65 85.14 84.62 84.10 83.58 83.06 82.54 82.02 81.50  
80.98  
80.46 88.82 99.28 112.72 130.63 155.72 154.68 153.64 152.61 151.57 150.53  
149.49 148.45 147.41 146.37 145.33 144.29 143.25 142.21 141.17 140.14  
139.10 138.06 137.02 135.98 134.94 133.90 132.86 131.82 130.78 129.74  
128.71 127.67 126.63 125.59 124.55 123.51 122.47 121.43 120.39 119.35  
118.31 117.27 116.24 115.20 114.16 113.12 112.08 111.04 110.00 108.96  
107.92 106.88 105.84 104.80 103.77 102.73 101.69 100.65 99.61 98.57 97.53  
96.49 95.45 94.41 93.37 92.34 91.30 90.26 89.22 88.18 87.14 86.10 85.06  
84.02 82.98]

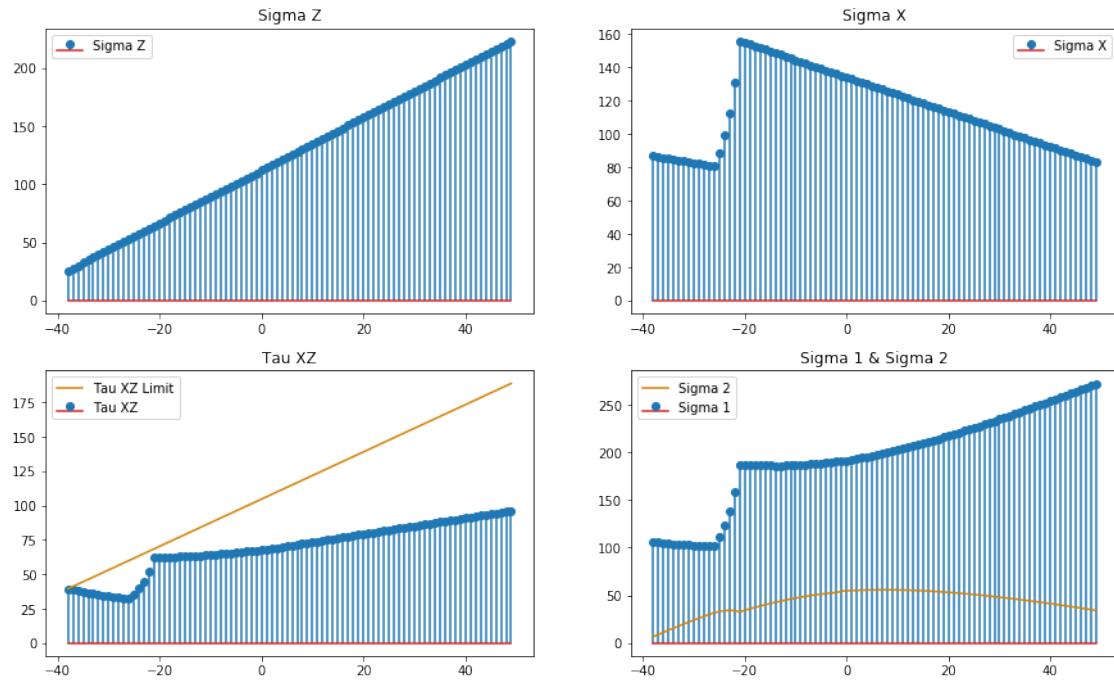
p\_a and p\_b = (105.2, 7.2000000000000003)  
tau xz,A = 38.98739744719502  
tau xz,B = 127.47565703385138  
delta Q = -19199.38555609435  
tau xz = [39.26 38.52 37.81 37.12 36.46 35.83 35.23 34.66 34.11 33.59 33.10

```

32.63
32.19 35.35 39.41 44.77 52.04 62.37 62.38 62.42 62.49 62.58 62.71 62.85
63.03 63.23 63.47 63.72 64.01 64.32 64.66 65.03 65.43 65.85 66.30 66.78
67.28 66.82 67.40 67.99 68.58 69.16 69.75 70.34 70.92 71.51 72.10 72.68
73.27 73.85 74.44 75.03 75.61 76.20 76.79 77.37 77.96 78.55 79.13 79.72
80.31 80.89 81.48 82.07 82.65 83.24 83.83 84.41 85.00 85.59 86.17 86.76
87.35 87.93 88.52 89.11 89.69 90.28 90.87 91.45 92.04 92.63 93.21 93.80
94.39 94.97 95.56 96.15]
tau xz limit = [39.36 41.07 42.79 44.51 46.22 47.94 49.65 51.37 53.08 54.80
56.51 58.23
59.95 61.66 63.38 65.09 66.81 68.52 70.24 71.95 73.67 75.38 77.10 78.82
80.53 82.25 83.96 85.68 87.39 89.11 90.82 92.54 94.26 95.97 97.69 99.40
101.12 102.83 104.55 106.26 107.98 109.69 111.41 113.13 114.84 116.56
118.27 119.99 121.70 123.42 125.13 126.85 128.57 130.28 132.00 133.71
135.43 137.14 138.86 140.57 142.29 144.00 145.72 147.44 149.15 150.87
152.58 154.30 156.01 157.73 159.44 161.16 162.88 164.59 166.31 168.02
169.74 171.45 173.17 174.88 176.60 178.32 180.03 181.75 183.46 185.18
186.89 188.61]
tau check = 0, if zero - everything OK

sigma 1 = [105.90 105.34 104.81 104.31 103.85 103.43 103.04 102.70 102.39 102.14
101.94 101.79 101.69 111.14 122.98 138.23 158.65 187.33 186.99 186.69
186.45 186.25 186.11 186.02 185.99 186.01 186.09 186.24 186.44 186.71
187.05 187.45 187.92 188.45 189.06 189.73 190.48 190.31 191.22 192.17
193.16 194.20 195.28 196.40 197.55 198.75 199.99 201.26 202.57 203.91
205.29 206.70 208.15 209.62 211.13 212.66 214.22 215.81 217.42 219.06
220.73 222.41 224.12 225.85 227.60 229.38 231.17 232.98 234.80 236.65
238.51 240.39 242.28 244.18 246.10 248.04 249.99 251.94 253.92 255.90
257.89 259.90 261.91 263.94 265.97 268.02 270.07 272.13]
sigma 2 = [6.43 8.74 11.02 13.27 15.49 17.66 19.80 21.90 23.95 25.95 27.91 29.81
31.66 32.84 33.73 34.19 33.97 32.64 34.22 35.74 37.22 38.65 40.03 41.35
42.61 43.82 44.97 46.06 47.09 48.05 48.95 49.78 50.54 51.24 51.87 52.42
52.91 54.31 54.64 54.92 55.16 55.35 55.51 55.62 55.69 55.73 55.73 55.69
55.61 55.50 55.35 55.17 54.96 54.72 54.45 54.15 53.82 53.46 53.08 52.67
52.24 51.79 51.31 50.81 50.29 49.75 49.19 48.62 48.02 47.41 46.78 46.14
45.48 44.81 44.12 43.42 42.70 41.97 41.24 40.48 39.72 38.95 38.17 37.37
36.57 35.76 34.94 34.11]
tau max = [49.73 48.30 46.89 45.52 44.18 42.88 41.62 40.40 39.22 38.09 37.01
35.99
35.02 39.15 44.62 52.02 62.34 77.34 76.38 75.47 74.61 73.80 73.04 72.34
71.69 71.09 70.56 70.09 69.68 69.33 69.05 68.84 68.69 68.61 68.60 68.66
68.78 68.00 68.29 68.62 69.00 69.42 69.88 70.39 70.93 71.51 72.13 72.79
73.48 74.21 74.97 75.77 76.59 77.45 78.34 79.26 80.20 81.17 82.17 83.19
84.24 85.31 86.41 87.52 88.66 89.81 90.99 92.18 93.39 94.62 95.86 97.12
98.40 99.69 100.99 102.31 103.64 104.99 106.34 107.71 109.09 110.47
111.87 113.28 114.70 116.13 117.57 119.01]

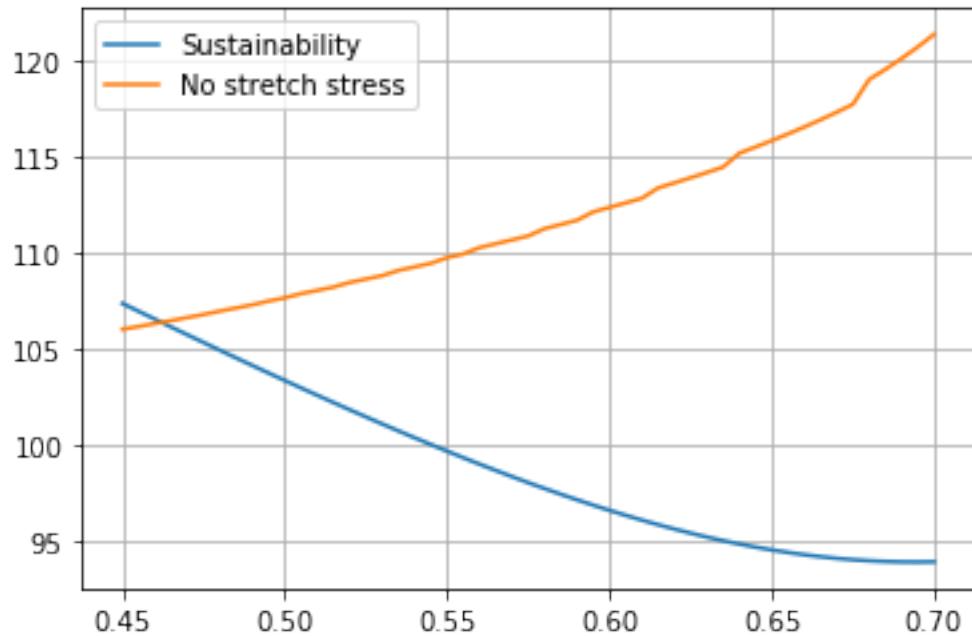
```



[6]: 0

0.3 Ht = 100 m

[7]: `plot_bn(Ht_plot=100, B_range=(60, 111), save_fig="Bn100")`



### 0.3.1 B=106.5, n=0.46

```
[8]: static_calc(Ht=100, B=106.5, n=0.46, plots=True, print_init=True, ▾
→print_results=True, output="tau_xz_check", save_fig="sigmatau100")
```

```
Ht = 100
B = 106.5
n = 0.46
d = 10
D = 20
b = 15.0
m = 0.605
dc = 1.62847272727273
F = 1242.5181758116341
x_A = 46.92044519383035
x_B = 59.57955480616965
Jy = 1310148.7771828296
W_A = 27922.769525535175
W_B = 21989.905454062904
```

#### Summary

```
Construction period
N = 170246.39811937205
M = -163640.8926458099
Working period
N = 173029.84257074876
Q = 105555.20967690594
M = 3017419.0863357643
```

```
Stretch criteria = -209741.2157701552
Sustainability criteria = -41.63016807087115
```

#### Stresses:

```
sigma z,A = 31.194363330905716
sigma z,B = 277.6273241219469
coordinates = [-46.92 -45.92 -44.92 -43.92 -42.92 -41.92 -40.92 -39.92 -38.92
-37.92
-36.92 -35.92 -34.92 -33.92 -32.92 -31.92 -30.92 -29.92 -28.92 -27.92
-26.92 -25.92 -24.92 -23.92 -22.92 -21.92 -20.92 -19.92 -18.92 -17.92
-16.92 -15.92 -14.92 -13.92 -12.92 -11.92 -10.92 -9.92 -8.92 -7.92 -6.92
-5.92 -4.92 -3.92 -2.92 -1.92 -0.92 0.08 1.08 2.08 3.08 4.08 5.08 6.08
7.08 8.08 9.08 10.08 11.08 12.08 13.08 14.08 15.08 16.08 17.08 18.08
19.08 20.08 21.08 22.08 23.08 24.08 25.08 26.08 27.08 28.08 29.08 30.08
31.08 32.08 33.08 34.08 35.08 36.08 37.08 38.08 39.08 40.08 41.08 42.08
43.08 44.08 45.08 46.08 47.08 48.08 49.08 50.08 51.08 52.08 53.08 54.08
```

```

55.08 56.08 57.08 58.08 59.08 60.08]
sigma z = [31.19 33.50 35.80 38.10 40.41 42.71 45.01 47.32 49.62 51.92 54.23
56.53
58.83 61.13 63.44 65.74 68.04 70.35 72.65 74.95 77.26 79.56 81.86 84.17
86.47 88.77 91.08 93.38 95.68 97.98 100.29 102.59 104.89 107.20 109.50
111.80 114.11 116.41 118.71 121.02 123.32 125.62 127.93 130.23 132.53
134.83 137.14 139.44 141.74 144.05 146.35 148.65 150.96 153.26 155.56
157.87 160.17 162.47 164.77 167.08 169.38 171.68 173.99 176.29 178.59
180.90 183.20 185.50 187.81 190.11 192.41 194.72 197.02 199.32 201.62
203.93 206.23 208.53 210.84 213.14 215.44 217.75 220.05 222.35 224.66
226.96 229.26 231.57 233.87 236.17 238.47 240.78 243.08 245.38 247.69
249.99 252.29 254.60 256.90 259.20 261.51 263.81 266.11 268.41 270.72
273.02 275.32 277.63]

sigma x,A = 105.30840728081965
sigma x,B = 106.18316131173562
sigma x = [105.80 105.31 104.82 104.33 103.84 103.35 102.86 102.37 101.88 101.39
100.90 100.41 99.92 99.43 98.94 98.45 108.84 121.83 138.54 160.81 191.99
191.01 190.03 189.05 188.07 187.09 186.11 185.13 184.15 183.17 182.19
181.21 180.23 179.25 178.27 177.28 176.30 175.32 174.34 173.36 172.38
171.40 170.42 169.44 168.46 167.48 166.50 165.52 164.54 163.56 162.58
161.60 160.61 159.63 158.65 157.67 156.69 155.71 154.73 153.75 152.77
151.79 150.81 149.83 148.85 147.87 146.89 145.91 144.92 143.94 142.96
141.98 141.00 140.02 139.04 138.06 137.08 136.10 135.12 134.14 133.16
132.18 131.20 130.22 129.24 128.25 127.27 126.29 125.31 124.33 123.35
122.37 121.39 120.41 119.43 118.45 117.47 116.49 115.51 114.53 113.55
112.57 111.58 110.60 109.62 108.64 107.66 106.68]

p_a and p_b = (125.2, 7.200000000000003)
tau xz,A = 43.24259286778337
tau xz,B = 163.60853109377788
delta Q = -27619.694534720067
tau xz = [43.45 42.83 42.23 41.66 41.11 40.57 40.06 39.57 39.10 38.66 38.23
37.82
37.44 37.08 36.73 36.41 40.16 44.95 51.22 59.72 71.75 71.91 72.10 72.31
72.54 72.79 73.07 73.36 73.68 74.01 74.37 74.75 75.15 75.57 76.01 76.47
76.95 77.46 77.98 78.53 79.10 79.69 80.30 80.93 81.58 82.25 82.15 82.88
83.62 84.35 85.08 85.82 86.55 87.29 88.02 88.76 89.49 90.23 90.96 91.70
92.43 93.17 93.90 94.64 95.37 96.11 96.84 97.57 98.31 99.04 99.78 100.51
101.25 101.98 102.72 103.45 104.19 104.92 105.66 106.39 107.13 107.86
108.60 109.33 110.06 110.80 111.53 112.27 113.00 113.74 114.47 115.21
115.94 116.68 117.41 118.15 118.88 119.62 120.35 121.09 121.82 122.55
123.29 124.02 124.76 125.49 126.23 126.96]
tau xz limit = [43.51 45.24 46.98 48.71 50.45 52.18 53.92 55.65 57.39 59.12
60.86 62.60
64.33 66.07 67.80 69.54 71.27 73.01 74.74 76.48 78.21 79.95 81.69 83.42
85.16 86.89 88.63 90.36 92.10 93.83 95.57 97.30 99.04 100.78 102.51
104.25 105.98 107.72 109.45 111.19 112.92 114.66 116.39 118.13 119.87

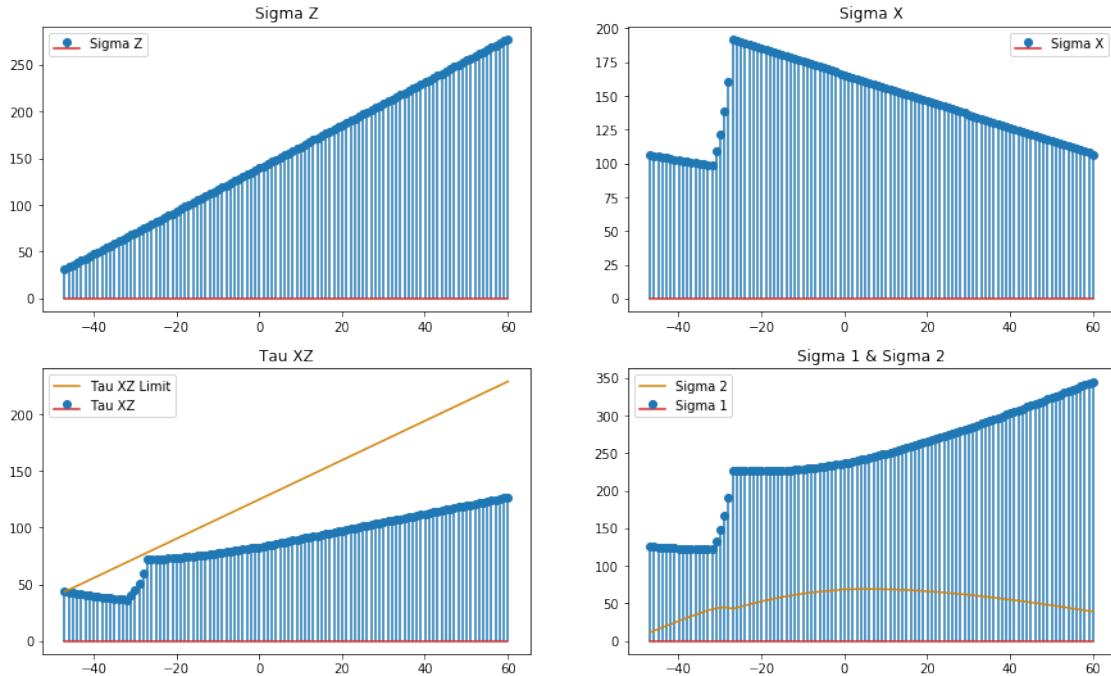
```

```

121.60 123.34 125.07 126.81 128.54 130.28 132.01 133.75 135.48 137.22
138.95 140.69 142.43 144.16 145.90 147.63 149.37 151.10 152.84 154.57
156.31 158.04 159.78 161.52 163.25 164.99 166.72 168.46 170.19 171.93
173.66 175.40 177.13 178.87 180.61 182.34 184.08 185.81 187.55 189.28
191.02 192.75 194.49 196.22 197.96 199.70 201.43 203.17 204.90 206.64
208.37 210.11 211.84 213.58 215.31 217.05 218.79 220.52 222.26 223.99
225.73 227.46 229.20]
tau check = 0, if zero - everything OK

sigma 1 = [125.76 125.29 124.85 124.43 124.04 123.68 123.35 123.05 122.78 122.55
122.35 122.20 122.08 122.01 121.98 122.01 133.49 147.89 166.50 191.43
226.49 226.27 226.08 225.94 225.83 225.77 225.75 225.78 225.85 225.97
226.13 226.35 226.62 226.93 227.30 227.73 228.20 228.74 229.32 229.97
230.66 231.42 232.23 233.10 234.03 235.01 235.27 236.38 237.53 238.71
239.93 241.19 242.47 243.79 245.14 246.53 247.94 249.38 250.85 252.35
253.88 255.43 257.01 258.62 260.24 261.90 263.57 265.27 266.99 268.72
270.48 272.26 274.06 275.88 277.71 279.56 281.43 283.31 285.21 287.13
289.06 291.00 292.95 294.92 296.91 298.90 300.91 302.92 304.95 306.99
309.04 311.10 313.17 315.25 317.34 319.44 321.55 323.66 325.78 327.91
330.05 332.19 334.35 336.51 338.67 340.84 343.02 345.21]
sigma 2 = [11.23 13.52 15.77 18.00 20.20 22.38 24.52 26.64 28.72 30.76 32.77
34.74
36.67 38.55 40.39 42.18 43.40 44.29 44.69 44.34 42.76 44.31 45.81 47.28
48.71 50.09 51.43 52.73 53.98 55.18 56.34 57.45 58.50 59.51 60.46 61.36
62.21 63.00 63.73 64.41 65.04 65.60 66.11 66.57 66.96 67.30 68.37 68.58
68.75 68.89 68.99 69.06 69.10 69.10 69.07 69.01 68.92 68.80 68.65 68.48
68.27 68.04 67.78 67.50 67.20 66.87 66.52 66.14 65.75 65.33 64.89 64.44
63.96 63.47 62.95 62.43 61.88 61.32 60.74 60.15 59.55 58.93 58.29 57.64
56.99 56.31 55.63 54.93 54.23 53.51 52.78 52.05 51.30 50.54 49.77 49.00
48.22 47.42 46.62 45.82 45.00 44.18 43.35 42.51 41.67 40.82 39.96 39.10]
tau max = [57.26 55.89 54.54 53.22 51.92 50.65 49.41 48.21 47.03 45.89 44.79
43.73
42.70 41.73 40.80 39.92 45.04 51.80 60.90 73.55 91.86 90.98 90.13 89.33
88.56 87.84 87.16 86.52 85.93 85.39 84.90 84.45 84.06 83.71 83.42 83.18
83.00 82.87 82.79 82.78 82.81 82.91 83.06 83.27 83.53 83.86 83.45 83.90
84.39 84.91 85.47 86.06 86.69 87.35 88.04 88.76 89.51 90.29 91.10 91.94
92.80 93.70 94.61 95.56 96.52 97.51 98.53 99.56 100.62 101.70 102.80
103.91 105.05 106.21 107.38 108.57 109.77 111.00 112.23 113.49 114.75
116.04 117.33 118.64 119.96 121.29 122.64 124.00 125.36 126.74 128.13
129.53 130.94 132.36 133.78 135.22 136.66 138.12 139.58 141.05 142.52
144.01 145.50 147.00 148.50 150.01 151.53 153.05]

```



[8]: 0

Ht = 60, 80, 100 B = 67.5, 86.2, 106.5 n = 0.57, 0.49, 0.46

```
[9]: def ht_plot(save_fig=None, _plots_transparent=PLOTS_TRANSPARENT):
    Ht_array = np.array([60, 80, 100])
    B_array = np.array([67.3, 86.2, 106.5])
    n_array = np.array([0.57, 0.49, 0.46])

    color2 = "#d3840c"

    fig, (ax1, ax2) = plt.subplots(figsize=(15,5), ncols=2, nrows=1)
    ax1.plot(Ht_array, B_array, label="B (Ht)")
    ax1.set_ylabel("B, meters")
    ax1.set_xlabel("Ht, meters")
    ax1.set_title("B (Ht) dependency")
    ax1.grid()
    ax1.legend()

    Ht_new = np.linspace(Ht_array.min(), Ht_array.max(), 300)
    spl = make_interp_spline(Ht_array, n_array, k=2)
    n_smooth = spl(Ht_new)

    ax2.plot(Ht_new, n_smooth, label="n (Ht)", color=color2)
    ax2.set_ylim(0.2, 0.7)
    ax2.set_ylabel("n")
    ax2.set_xlabel("Ht, meters")
```

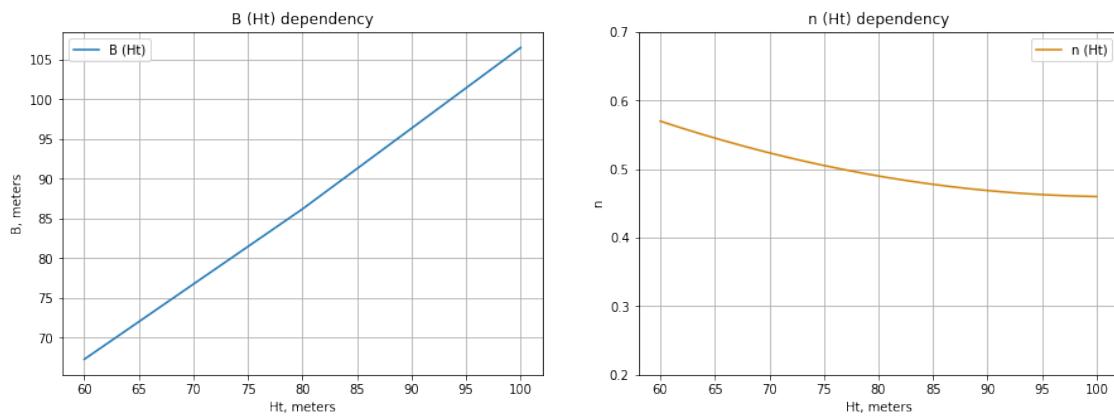
```

ax2.set_title("n (Ht) dependency")
ax2.grid()
ax2.legend()

if save_fig != None:
    if _plots_transparent == True:
        save_fig = "t_" + save_fig
    plt.savefig(save_fig + ".png", dpi=300, transparent=_plots_transparent)

ht_plot(save_fig="BnHt")

```



[ ]: