

Hydraulic Stability Calculator for Breakwater Armor Units

Concrete Cubes & Antifer Blocks

1. Abstract

This repository contains a comprehensive computational suite designed for the hydraulic design of coastal breakwater armor layers. The software implements advanced empirical formulae to dimension artificial concrete units, specifically **Simple Cubes** (Van der Meer) and **Antifer Blocks** (Chegini & Aghtouman).

A distinguishing feature of this tool is its **Iso-Geometric Design Philosophy** for the breakwater head. Rather than increasing the nominal diameter (D_n) of armor units at the roundhead—which necessitates different casting moulds, storage logistics, and crane requirements—this calculator solves for the required **increase in concrete density** (ρ_c). This allows the Trunk and the Head to be constructed using geometrically identical units (same mould), optimizing construction efficiency while meeting strict safety factors via material density adjustments.

2. Theoretical Framework & Methodology

The software calculates design parameters based on a synthesis of hydraulic model test data and established coastal engineering standards.

2.1 Wave Mechanics & Geometric Parameters

Before determining armor stability, the software computes the fundamental wave and structure properties:

Deep Water Wavelength (L_0): Calculated based on the mean wave period (T_m) using linear wave theory approximation for deep water:

$$L_0 = \frac{gT_m^2}{2\pi}$$

Wave Steepness (s_{0m}): A critical parameter influencing breaker type and stability:

$$s_{0m} = \frac{H_s}{L_0}$$

Number of Waves (N_z): The total number of waves attacking the structure during the design storm duration (t in hours):

$$N_z = \frac{t \times 3600}{T_m}$$

Relative Buoyant Density (Δ): The dimensionless density of the concrete relative to seawater:

$$\Delta = \frac{\rho_c}{\rho_w} - 1$$

2.2 Trunk Stability: Empirical Power Laws

The core of the calculator utilizes power-law formulas derived from extensive hydraulic model testing. The Stability Number (N_s) is the governing dimensionless parameter.

General Stability Formula:

$$N_s = \left(k_1 \frac{N_{od}^{k_2}}{N_z^{k_3}} + k_4 \right) s_{0m}^{-k_5}$$

Where: * N_{od} is the Damage Number (normalized damage level). * $k_{1..5}$ are empirical coefficients specific to the block type and slope.

Nominal Diameter (D_n): Once N_s is determined, the required characteristic size of the block is:

$$D_n = \frac{H_s}{\Delta \cdot N_s}$$

Armor Unit Weight (W):

$$W = \rho_c \cdot D_n^3$$

Empirical Coefficients (k) The software utilizes the following database of coefficients:

Method	Block		k_1	k_2	k_3	k_4	k_5
	Type	Slope					
Van der Meer (1988)	Simple Cubes	2.0:1	6.7	0.4	0.3	1.0	0.1
Van der Meer (1988)	Simple Cubes	1.5:1	6.7	0.4	0.3	1.0	0.1
Chegini (2006)	Antifer	2.0:1	6.138	0.443	0.276	1.164	0.07
Chegini (2006)	Antifer	1.5:1	6.951	0.443	0.291	1.082	0.082

Note: For Van der Meer Cubes (Slope 2.0:1), the script applies a specific calibration factor adjustment to the result found in the source code:

$$N_{s,adjusted} = N_s \times \left(\frac{2.0}{1.5} \right)^{1/3}$$

2.3 The Head Design Strategy: Van der Meer to Hudson Transfer

The breakwater head is subjected to severe 3D turbulence and wave breaking, significantly higher than the trunk. Standard practice suggests increasing the block weight. However, this tool uses a “Transfer Function” approach to maintain constant geometry.

The Logic: 1. **Calculate Equivalent Hudson K_D :** We first determine what the Hudson Stability Coefficient (K_D) would be for the calculated Trunk armor.

$$K_{D,trunk} = \frac{\rho_c H_s^3}{W \Delta^3 \cot \alpha}$$

2. **Apply Safety Ratio (R):** We enforce that the Head must be 1.5 times more stable than the Trunk (standard engineering judgment).

$$K_{D,head} = \frac{K_{D,trunk}}{1.5}$$

3. **Solve for Density Increase:** In the Hudson formula, stability is proportional to Δ^3 . To maintain the same weight W and size D_n while increasing stability, we must increase Δ .

$$\Delta_{head} = \Delta_{trunk} \cdot (1.5)^{1/3}$$

4. **Result:** The software outputs a required **High Density Concrete** ($\rho_{c,head}$) for the head units.

$$\rho_{c,head} = \rho_w (\Delta_{head} + 1)$$

2.4 Underlayer Geotechnical Design (EN 13383)

The software automatically sizes the underlayer rock based on the **EN 13383** standard European rock gradings.

1. **Target Weight:** $W_{ul} \approx W_{armor}/10$
2. **Selection Algorithm:** The code iterates through standard grading classes (e.g., LMA 60-300kg, HMA 1-3 ton) and selects the class where the mean weight (W_{50}) is closest to the target.
3. **Thickness (r_2):** Calculated based on a double layer of the selected rock size.

3. Repository Contents & Compilation

This repository includes three implementation variations of the calculator to suit different engineering workflows.

3.1 Python Script (`breakwater-cubes-calculator.py`)

A rapid prototyping script ideal for quick checks and education. * **Dependencies:** Python 3.x (Standard libraries: `math`, `sys`, `os`). * **Execution:** `bash python3 breakwater-cubes-calculator.py`

3.2 C++ CLI (`breakwater_calculator_cli.cpp`)

A high-performance command-line tool designed for batch processing or integration into pipelines. It utilizes static linking for portability.

- **Compilation (GCC/MinGW):** `bash g++ -O3 -march=native -std=c++17 -Wall -Wextra \ -static -static-libgcc -static-libstdc++ \ -o breakwater_calculator_cli breakwater_calculator_cli.cpp`
- **Usage (Interactive):** `./breakwater_calculator_cli`
- **Usage (Arguments):** `bash ./breakwater_calculator_cli [Hs] [Tm] [Duration] [Nod] [Wc] [FormulaID] # Example: ./breakwater_calculator_cli 10.5 13.0 12.0 1.0 24.0 1`

3.3 C++ GUI (`breakwater_calculator_gui.cpp`)

A standalone native Windows application using the Win32 API. It provides a visual interface for inputting parameters and viewing generated reports.

- **Compilation (MinGW on Windows):** `bash g++ -O3 -march=native -std=c++17 -municode \ breakwater_calculator_gui.cpp -o breakwater_calculator_gui \ -mwindows -static -static-libgcc -static-libstdc++`
- **Features:**
 - Form-based inputs with default values.
 - Dropdown selection for Formula/Block type.
 - Scrollable report output window.
 - Automatic logging to `output.txt`.

4. Input Parameter Definitions

Parameter	Symbol	Unit	Description
Sig. Wave Height	H_s	meters	The average height of the highest 1/3 of waves at the toe of the structure.
Mean Period	T_m	seconds	The average wave period.
Damage Number	N_{od}	-	Allowable damage level. $N_{od} = 0$: No damage $N_{od} = 0.5$: Start of damage $N_{od} = 2.0$: Severe damage
Storm Duration	t	hours	Duration of the peak design storm event (determines N_z).
Concrete Density	ρ_c	kN/m ³	Specific weight of the concrete used in the Trunk.
Formula ID	-	-	Selects between Van der Meer (Cubes) or Chegini (Antifer) and slope variations.

5. Output Report Explanation

The software generates a detailed technical text report (printed to stdout or `output.txt`). Key sections include:

1. **Intermediate Parameters:** L_0 , s_{0m} , N_z , and the calculated Stability Number (N_s).
2. **Trunk Results:** The required mass (M), nominal diameter (D_n), and

- specific dimensions (A, B, H) for the block.
3. **Head Results:** The derived required concrete density ($\rho_{c,head}$) to allow the use of trunk-sized blocks at the head.
 4. **Underlayer:** The specific EN 13383 grading selected (e.g., “EN 13383 HMA 1-3 ton”) and its layer thickness (r_2).
-

6. Bibliographic References

1. **Van der Meer, J.W. (1988).** *Rock Slopes and Gravel Beaches Under Wave Attack*. Doctoral Thesis, Delft University of Technology.
 2. **Van der Meer, J.W. (1988).** “Stability of Cubes, Tetrapods and Acropode.” *Proceedings of the Conference Breakwaters '88*, Eastbourne, Thomas Telford.
 3. **Chegini, V., & Aghtouman, P. (2006).** “An Investigation on the Stability of Rubble Mound Breakwaters with Armour Layers of Antifer Cubes.” *Journal of Marine Engineering*.
 4. **USACE (2006).** *Coastal Engineering Manual (CEM)*, Chapter VI-5.
 5. **CEN (2002).** *EN 13383-1: Armourstone - Part 1: Specification*.
-

7. License & Disclaimer

License: Open Source (MIT or equivalent).

Disclaimer: This software is an engineering aid and does not replace physical model testing. The authors assume no liability for the structural failure of breakwaters designed using these codes. Engineering judgment must be exercised, particularly regarding the specific hydraulic boundary conditions and material quality.