

The Sverdrup-Munk-Bretschneider (SMB) Wave Prediction Framework: A Theoretical, Historical, and Computational Manual

1. Introduction and Theoretical Scope

The prediction of wind-generated surface gravity waves constitutes one of the fundamental problems in geophysical fluid dynamics and coastal engineering. The primary objective is to forecast the statistical properties of the sea state—specifically the **Significant Wave Height (H_s)** and the **Significant Wave Period (T_s)**—given a set of meteorological forcing parameters (wind speed, duration) and geometric constraints (fetch length, water depth).

1.1 The Duality of Wave Modeling

In modern oceanography, wave prediction is generally approached through two distinct paradigms:

1. **Spectral (Phase-Averaged) Models:** Third-generation models like **SWAN** (Simulating Waves Nearshore) and **WAVEWATCH III** solve the **Action Balance Equation**:

$$\frac{\partial N}{\partial t} + \nabla \cdot (\vec{c}_g N) = \frac{S}{\sigma}$$

Where N is the wave action density, \vec{c}_g is the group velocity vector, and S represents source terms (wind input, quadruplet interactions, whitecapping). While physically rigorous, these models require extensive computational grids and detailed boundary conditions.

2. **Parametric (Empirical) Models:** This software suite implements the parametric approach. Instead of solving the partial differential equation for the entire spectrum, parametric models rely on **Similarity Laws**. They assume that the wave spectrum, under constant wind forcing, maintains a self-similar shape (typically JONSWAP or Pierson-Moskowitz). This assumption allows the state of the sea to be described by simple non-dimensional relationships between total energy, peak frequency, and fetch.

1.2 The Scope of this Software

This repository provides a high-fidelity implementation of the **Sverdrup-Munk-Bretschneider (SMB)** method. Critically, it discards the discontinuous equations of the 1984 *Shore Protection Manual* (SPM) in favor of the **Hurdle and Stive (1989) Unified Formulation**.

The Hurdle and Stive revision is mathematically significant because it ensures **C^1 continuity** (smoothness of both the function and its derivative) across the transition from deep water to shallow water. This prevents the generation of non-physical artifacts—such as sudden jumps in predicted wave height—that occur when using legacy lookup tables or piecewise functions.

2. Foundations of Ocean Wave Science: The Sverdrup-Munk Era

To understand the mechanics of the SMB method, one must examine the foundational physics established during the urgency of World War II.

2.1 The Energy Budget Hypothesis (1947)

Before 1947, wave forecasting was based on empirical correlations (e.g., Stevenson's formula). **Harald Sverdrup and Walter Munk** revolutionized the field by introducing the **Energy Budget Hypothesis**. They posited that the growth of waves is not merely kinematic but is governed by the conservation of energy flux.

They formulated the governing differential equation for the total wave energy E :

$$\frac{dE}{dt} + \frac{E}{C} \frac{dC}{dt} = R_{Total} - D_{iss}$$

Where:

- $E = \frac{1}{8} \rho g H^2$: The mean energy per unit surface area.
- C : The phase velocity of the significant wave.
- R_{Total} : The net energy transfer from the atmosphere to the sea.
- D_{iss} : Energy dissipation (which they assumed negligible for growing waves until breaking).

2.2 Atmospheric Energy Transfer Mechanisms

Sverdrup and Munk identified two distinct physical mechanisms for wind-to-wave energy transfer, a duality that remains central to fluid dynamics today:

1. **Normal Pressure (R_N)**: This mechanism relies on **Form Drag** (or the sheltering effect). As wind blows over a wavy surface, the airflow separates. This creates a zone of high static pressure on the windward face of the crest and a zone of low pressure (suction) on the leeward face.
 - *Physics*: The pressure vector is phase-shifted relative to the surface elevation.

- *Effectiveness:* This is the dominant growth mechanism for young, steep waves where the wind speed U is much greater than the wave phase speed C ($U \gg C$).
2. **Tangential Stress (R_T):** This represents **Skin Friction**. It is the transfer of momentum via viscous shear stress at the air-sea interface.
- *Physics:* The wind "rubs" against the water surface, dragging particles forward.
 - *Effectiveness:* This mechanism operates even when waves are moving nearly as fast as the wind, but it is generally less efficient than form drag for rapid growth.

2.3 The Definition of Significant Wave Height

The sea surface is a stochastic, multi-scale phenomenon. To make it tractable for engineering, Sverdrup and Munk defined the **Significant Wave Height** ($H_{1/3}$).

- **Statistical Definition:** The average height of the highest one-third of waves in a zero-upcrossing record.
- **Spectral Definition:** In deep water, assuming a narrow-band spectrum, wave heights follow a **Rayleigh Distribution**. Under this assumption:

$$H_{1/3} \approx 4\sqrt{m_0}$$

Where m_0 is the variance (total energy) of the sea surface elevation.

3. The Evolution of Parametric Forecasting (1950-1984)

The transition from Sverdrup and Munk's theoretical basis to standard engineering curves involved decades of data collection and refinement, culminating in the discontinuous standardization of the 1980s.

3.1 Bretschneider's Fully Developed Sea (FDS)

C.L. Bretschneider (1958) utilized vast datasets from weather ships to define the upper limits of wave growth. He formalized the concept of the **Fully Developed Sea**.

- **Physical Concept:** FDS is a dynamic equilibrium where the energy input from the wind (R_{Total}) is exactly balanced by the energy dissipation (D_{iss}) due to whitecapping and turbulence.
- **Implication:** Once a sea is fully developed, neither an increase in fetch (distance) nor an increase in duration (time) will produce larger waves. The wave field is saturated.

Bretschneider established the dimensionless asymptotic limit for wave height as:

$$\frac{gH_{1/3}}{U^2} \approx 0.282$$

(Note: This coefficient was later revised to 0.25 by Hurdle & Stive to correct for instrument bias in early data).

3.2 The Shore Protection Manual (1984) Standardization Limitations

The 1984 edition of the *Shore Protection Manual* (SPM) became the global standard, but it introduced two critical methodological inconsistencies that this software explicitly corrects.

3.2.1 The Adjusted Wind Speed (U_A) Discrepancy The SPM attempted to account for the non-linear aerodynamics of the sea surface by defining an "Adjusted Wind Speed" U_A (or wind stress factor):

$$U_A = 0.71 \cdot (U_{10})^{1.23}$$

- **The Original Limitation (SPM 1984):** While physically motivated to represent Friction Velocity (u_*), this factor was applied to equations originally calibrated against different wind parameters. As noted by Bishop, Donelan, and Kahma (1992), this led to "double-counting" the non-linearity, causing a systematic **overprediction** of wave heights by 10-20% in high wind conditions.
- **The Hurdle & Stive (1989) Correction:** To fix this inconsistency without abandoning the U_A methodology, Hurdle & Stive recalibrated the empirical coefficients. They **lowered** the wave height coefficient from 0.283 to **0.25** and adjusted the period coefficient to **8.3**.
- **Operational Requirement:** Therefore, U_A **must still be calculated** when using this software. The lower coefficients in the Hurdle & Stive formulas are specifically calibrated to compensate for the adjusted wind speed, ensuring the final prediction is accurate.

3.2.2 The Mathematical Discontinuity The SPM provided three separate sets of equations for Deep, Intermediate, and Shallow water.

- **Deep Water:** Dependent only on Wind and Fetch.
- **Shallow Water:** Heavily dependent on Depth (d).
- **The Singularity:** At the transition depth (where the deep and shallow curves should meet), the equations often yielded different values.
 - *Example:* Calculating for a 50km fetch at 20m/s wind might yield $H_s = 4.2m$ using the deep water formula, but $H_s = 3.8m$ using the shallow water formula at a depth of 200m (which should be effectively deep).

- *Impact:* This step-change forced engineers to arbitrarily smooth data or choose a “conservative” (often overly expensive) design value.

The **Hurdle and Stive (1989)** formulation used in this software resolves this by using a single, unified equation that asymptotically approaches the deep water limit as $d \rightarrow \infty$ and the shallow water limit as $d \rightarrow 0$.

4. The Hurdle and Stive (1989) Unification: Theoretical Manual

The core physics engine of this software is built upon the landmark paper **”Revision of SPM 1984 wave hindcast model to avoid inconsistencies in engineering applications”** by D.P. Hurdle and R.J.H. Stive (Delft Hydraulics), published in *Coastal Engineering* (1989).

This section details the mathematical architecture of their formulation, which was developed to resolve the **critical numerical and physical instabilities** present in the U.S. Army Corps of Engineers' 1984 *Shore Protection Manual* (SPM).

4.1 Historical Necessity: The ”Crisis of Inconsistency” (1984–1989)

Prior to 1989, coastal engineers relied on the SPM (1984) as the global standard. However, the manual contained **fundamental mathematical inconsistencies** that became apparent when engineers attempted to digitize the curves for computer models.

Hurdle and Stive (1989) clearly identified those inconsistencies in the SPM 1984 formulation:

1. **The ”Transition Gap” (Deep Water Inconsistency):** The SPM provided two separate sets of equations for deep water: one for ”fetch-limited” growing seas and another for ”fully developed” seas. **These two curves did not mathematically intersect.** At the transition point, the predicted wave height would suddenly ”jump” or exhibit a discontinuity in slope.
2. **The ”Asymptotic Failure” (Shallow Water Inconsistency):** The SPM's shallow water equations were derived independently from the deep water equations. As a result, if one calculated wave growth in shallow water and mathematically increased the depth to infinity ($d \rightarrow \infty$), the result **did not converge** to the deep water prediction.

The Engineering Consequence: These artifacts meant that a software program implementing the SPM 1984 literally **could not ensure a smooth, continuous solution**. A negligible change in input (e.g., increasing depth by 1 cm) could trigger a switch between equation sets, causing a non-physical jump in design wave height.

4.2 Theoretical Derivation: The Hyperbolic Tangent Unification

To solve these singularities, Hurdle and Stive proposed a **Unified Formulation**. Instead of using piecewise functions (different equations for different depths), they derived a single, continuous function valid for **all water depths**, from the surf zone to the abyssal ocean.

They achieved this using the **Hyperbolic Tangent (tanh) Transformation**. The $\tanh(x)$ function is the "Swiss Army Knife" of coastal engineering because of its unique asymptotic properties:

- **The Deep Water Asymptote:** $\lim_{x \rightarrow \infty} \tanh(x) = 1$. This allows the equation to mathematically "ignore" the depth term when water is deep.
- **The Shallow Water Asymptote:** $\lim_{x \rightarrow 0} \tanh(x) \approx x$. This forces the equation to become linearly dependent on depth as water becomes shallow.

By nesting these functions, Hurdle and Stive ensured **C^1 Continuity**: The prediction surface is smooth in both value (C^0) and slope (C^1) everywhere. **There are no "kinks," no jumps, and no regime-switching artifacts.**

4.3 The Unified Physics Engine: Detailed Equations

The software solves the following dimensionless equations. These formulas represent the corrected, unified physics derived by Hurdle & Stive.

4.3.1 Dimensionless Parameters The system is normalized using the **Adjusted Wind Speed (U_A)** and Gravitational Acceleration (g).

- **Dimensionless Fetch (\hat{F}):** $\hat{F} = \frac{gF}{U_A^2}$
- **Dimensionless Depth (\hat{d}):** $\hat{d} = \frac{gd}{U_A^2}$

(Note: U_A must be calculated first as $0.71U_{10}^{1.23}$. Using raw U_{10} here will result in significant errors.)

4.3.2 Significant Wave Height (H_s) Formula The unified equation for energy (height) is:

$$\frac{gH_s}{U_A^2} = 0.25 \cdot \tanh \left[\underbrace{0.6 \cdot \hat{d}^{0.75}}_{\text{Depth Limit Term } K_{d1}} \right] \cdot \tanh^{0.5} \left[\frac{4.3 \times 10^{-5} \cdot \hat{F}}{\tanh^2(K_{d1})} \right]$$

Detailed Component Analysis:

- **The Leading Coefficient (0.25):** This is the **Asymptotic Stability Limit**. It represents the maximum dimensionless wave energy in a fully developed sea. **Important:** The SPM (1984) used 0.283 here. Hurdle & Stive **recalibrated** this down to 0.25 after finding that the SPM overpredicted wave heights by ~12%.

- **The Depth Limit Term (K_{d1}):** This inner tanh function acts as a "friction valve."
 - In **Deep Water**, $\tanh(K_{d1}) \rightarrow 1$. The depth term vanishes.
 - In **Shallow Water**, $\tanh(K_{d1})$ becomes small, reducing the effective fetch and capping the maximum wave height.

4.3.3 Significant Wave Period (T_s) Formula The unified equation for dispersion (period) is:

$$\frac{gT_s}{U_A} = 8.3 \cdot \tanh \left[\underbrace{\frac{0.76 \cdot \hat{d}^{0.375}}{K_{d2}}}_{\text{Depth Limit Term } K_{d2}} \right] \cdot \tanh^{1/3} \left[\frac{4.1 \times 10^{-5} \cdot \hat{F}}{\tanh^3(K_{d2})} \right]$$

Detailed Component Analysis:

- **The Leading Coefficient (8.3):** This represents the maximum wave period (spectral peak) in a fully developed sea. **Important:** The SPM (1984) used 7.54. Hurdle & Stive **increased** this to 8.3. This is a critical safety correction; under-predicting the wave period leads to underestimating the wavelength and orbital velocity.

4.4 Duration Limitation: The "Effective Fetch" Method

A common source of error in legacy models was the treatment of **Duration-Limited** conditions (i.e., when the wind doesn't blow long enough for waves to reach the end of the fetch).

The Legacy Error: Old models used a separate set of "Time-Growth" equations. These often violated kinematic principles, implying that waves traveled faster than their own group velocity in shallow water.

The Hurdle & Stive Solution (Implemented in `calculator_gui.cpp`): This software uses the **Effective Fetch** inversion method to ensure strict kinematic consistency.

1. **Calculate Minimum Duration (t_{min}):** Using the rigorous power law derived from the integration of group velocity:

$$t_{min} = \frac{65.9 \cdot U_A}{g} \cdot \hat{F}^{2/3}$$

2. **Check Condition:** If the actual storm duration (t_{act}) is less than t_{min} , the system is **Duration Limited**.
3. **Invert for Effective Fetch:** We solve for the "Effective Fetch" (F_{eff}) that would have produced the current state if the wind had blown forever.

$$F_{eff} = \left(\frac{t_{act} \cdot g}{65.9 \cdot U_A} \right)^{1.5} \cdot \frac{U_A^2}{g}$$

4. **Substitute:** This F_{eff} is then substituted back into the unified H_s and T_s equations (Section 4.3) to generate the final wave parameters.

Significance: This approach guarantees that the duration-limited growth curve follows **exactly the same trajectory** as the fetch-limited curve, preventing mathematical hysteresis.

4.5 Controlling Factor Logic (Algorithm)

To determine the final design sea state, the software calculates potential wave growth under both assumptions and selects the physically limiting factor:

1. Calculate $H_{s,fetch}$ (Assuming infinite duration).
2. Calculate $H_{s,duration}$ (Assuming infinite fetch).
3. **Determine Controlling Condition:**

$$H_{s,final} = \min(H_{s,fetch}, H_{s,duration})$$

This logic ensures the model is physically conservative: a wave cannot grow larger than the fetch allows, nor larger than the duration allows.

5. Computational Numerical Methods

5.1 Exact Linear Dispersion (Newton-Raphson)

Once T_s is determined, the software calculates the Wavelength (L). This requires solving the **Linear Dispersion Relation**:

$$L = \frac{gT^2}{2\pi} \tanh\left(\frac{2\pi d}{L}\right)$$

This is a **transcendental equation**. To ensure both accuracy and computational efficiency, this software implements an **exact numerical solution** using the **Newton-Raphson method**, initialized by a high-precision explicit approximation.

Algorithm Steps:

1. **Define parameters:** Dimensionless deep-water wavenumber $k_0d = \frac{\omega^2 d}{g}$, where $\omega = \frac{2\pi}{T}$.
2. **Initialize (Carvalho, 2006):** Generate a high-quality initial guess (kh_{init}) to ensure rapid convergence:

$$kh_{init} = \frac{k_0d}{\tanh\left(\sqrt{k_0d} \cdot (6/5)^{(k_0d)}\right)}$$

3. **Define Root Function:** $f(kh) = kh \tanh(kh) - k_0d$.

4. **Derive Gradient:** $f'(kh) = \tanh(kh) + kh \cdot \operatorname{sech}^2(kh)$.
5. **Iterate:** Apply the update rule until the error $< 10^{-15}$:

$$kh_{n+1} = kh_n - \frac{f(kh_n)}{f'(kh_n)}$$

This hybrid approach combines the speed of an explicit approximation with the precision of an iterative solver, ensuring robust performance across all depth regimes ($\pi/10 < kh < \pi$).

5.2 The Miche Stability Criterion (Wave Breaking)

To ensure the predicted sea state is physically sustainable, the model applies the **Miche Criterion** (1944). This is the fundamental limit governing wave steepness; beyond this threshold, the wave crest becomes unstable and breaks (whitecapping in deep water or plunging/spilling in shallow water).

1. The Physical Principle: Kinematic Instability Waves break when the horizontal velocity of the water particles (u) at the wave crest exceeds the speed at which the wave form is traveling (Phase Celerity, C).

- If $u > C$, the water particles physically outrun the wave form, causing the crest to collapse forward.
- In deep water, this corresponds to a limiting crest angle of **120°** (Stokes' Limit).

2. The Mathematical Formulation Miche derived a unified equation that defines this limiting steepness (H/L) as a function of water depth:

$$\left(\frac{H}{L}\right)_{max} = 0.142 \cdot \tanh(kd)$$

Where:

- $0.142 \approx 1/7$: The theoretical maximum steepness in deep water.
- $\tanh(kd)$: The depth-attenuation factor that reduces the allowable steepness as water gets shallower.

3. Asymptotic Limits This single equation governs breaking in both major regimes:

- **Deep Water Limit ($kd > \pi$)**: The $\tanh(kd)$ term approaches 1.0. The limit simplifies to $H/L \leq 0.142$ (approx. 1/7).
- **Shallow Water Limit ($kd < \pi/10$)**: The $\tanh(kd)$ term approaches kd . The equation simplifies to linear depth-limited breaking:

$$\frac{H}{L} \approx 0.142(kd) \implies H \approx 0.142 \cdot kd \cdot L \approx 0.89d$$

(Note: While empirical indices often range from 0.78 to 1.0, Miche's theoretical derivation yields ≈ 0.89).

4. Software Implementation After calculating H_s and L , the software calculates the current steepness ratio. If the predicted steepness exceeds the Miche limit, the software flags the sea state as "**BREAKING / UNSTABLE**", indicating that the wind input is generating energy faster than the water surface can sustain it without dissipating via breaking.

6. Software Reference: Script Descriptions and Usage

This section provides a detailed breakdown of every script in the repository, explaining its purpose, internal logic, and usage.

6.1 calculator.py: The Interactive Core

This is the primary Python implementation of the physics engine.

- **Purpose:** Provides a text-based, interactive environment for single-point calculations. It is ideal for rapid sensitivity analysis (e.g., "What if the fetch increases by 5km?").
- **Class Tee:**
 - **Description:** A utility class that mimics the Unix `tee` command.
 - **Functionality:** It overrides `sys.stdout`. Every `print()` statement in the script is simultaneously written to the console and to a log file named `report.txt`. This creates an automatic audit trail of all calculations performed during a session.
- **Logic Flow:**
 1. **Input:** Prompts user for Wind Speed (U_{10}), Fetch, Duration, and Depth.
 2. **Conversion:** Calculates Adjusted Wind Speed (U_A).
 3. **Fetch Calculation:** Computes the potential wave height based on the physical fetch.
 4. **Duration Calculation:** Computes the **Effective Fetch** derived from the storm duration (using the inverted Hurdle & Stive power law) and calculates the resulting potential wave height.
 5. **Minimization:** Selects the limiting condition (Fetch vs. Duration) based on which produces the smaller wave height.
- **Usage:** Run `python calculator.py` in a terminal.

6.2 calculator_gui.cpp: The High-Performance Application

This is the C++ implementation designed for Windows.

- **Purpose:** A graphical user interface (GUI) tool for end-users who prefer not to use the command line.
- **Class SMBEngine:**
 - **Description:** A stateless physics engine class.

- **Methods:** Uses static methods like `calculate_adjusted_wind_speed`, `calculate_deep_water`, and `solve_dispersion`. Being stateless ensures thread safety.
- **Class StringLogger:** Handles the formatting of the output report, ensuring decimal precision and unit alignment.
- **Compilation:** Requires a C++17 compliant compiler (GCC/MinGW) with static linking flags: `g++ -O3 -std=c++17 -static -static-libgcc -static-libstdc++ -o calculator_gui.exe calculator_gui.cpp -mwindows -lgdi32`.

6.3 `calculator.ipynb`: The Interactive Notebook

This is the Jupyter Notebook port of the C++ application.

- **Purpose:** Allows users to inspect the physics engine step-by-step and run calculations without compiling code.
- **Structure:** The code is modularized into independent cells for educational clarity:
 - **Core Logic:** Contains the `SMBEngine` class, which is a direct line-by-line port of the C++ physics engine to Python.
 - **Logging:** Implements the `StringLogger` class to replicate the exact text report format found in the GUI.
 - **Execution:** The final cell provides an interactive input prompt. It includes intelligent defaults (Lake Garda test case) that trigger if the user simply presses **ENTER**.
- **Usage:** Open in Jupyter Lab, Jupyter Notebook, or VS Code and select **Run All Cells**.

6.4 `smb_chart_deep.py` and `smb_chart_10m.py`

These scripts generate engineering contour charts.

- **Libraries:** `numpy` (for mesh generation), `matplotlib` (for plotting).
- **Logic:**
 1. **Grid Generation:** Creates a meshgrid of Wind Speed (0-40 m/s) and Fetch (0-200 km).
 2. **Vectorized Calculation:** Applies the SMB formulas to the entire grid simultaneously to compute matrices for H_s , T_s , and t_{min} .
 3. **Contouring:** Uses `plt.contour` to draw isolines:
 - **Solid Black:** Significant Wave Height (H_s).
 - **Dashed Black:** Significant Wave Period (T_s).
 - **Dotted Black:** Minimum Duration (t_{min}).
- **Distinction:** `smb_chart_deep.py` assumes infinite depth ($\tanh(K_d) = 1$), while `smb_chart_10m.py` applies the shallow water damping functions for a fixed depth of 10 meters.

6.5 `smb-nomogram-deep.py` and `smb-nomogram-shallow.py`

These scripts generate professional alignment charts (Nomograms).

- **Libraries:** `pynomo` (`nomogen`), `pyx` (PostScript rendering).
- **Methodology (Type 9 Nomogram):**
 - These scripts solve the alignment problem for the equation $f(u) + g(v) = h(w)$.
 - They define the functions for Wind, Fetch, and Wave Height/Period.
 - The `nomogen` library numerically optimizes the position and curvature of the scales so that a straight line drawn across the three axes connects mathematically consistent values.

6.6 `tables.py`

This script generates a printable PDF lookup table.

- **Libraries:** `reportlab` (PDF generation).
- **Logic:**
 - Iterates through three nested loops:
 1. **Depths:** [5m, 10m, 25m, 50m, Deep].
 2. **Wind Speeds:** [10, 15, 20, 25, 30 m/s].
 3. **Fetches:** [5, 10, 25, 50 km].
 - Calculates the full wave state (H_s, T_s, t_{min}) for every combination.
 - Formats the data into a grid and draws it onto a PDF canvas, creating a multi-page reference document "tables.pdf".

7. Practical Example: Lago di Garda Case Study

To demonstrate the application of the Hurdle & Stive (1989) unified model, we present a practical hindcast simulation for **Lago di Garda**, Italy. As the largest lake in Italy, Lago di Garda represents an ideal test case for finite-depth physics due to its complex dual morphology and susceptibility to high-energy wind events.

- **Geographic Context:** Situated at the interface between the **Dolomiti** (Dolomites) and the **Pianura Padana** (Po Valley), the lake acts as a transition zone between Alpine and flatland weather systems.
- **Morphology:** The lake basin exhibits a distinctive "axe" shape divided into two distinct sectors:
 - **Alto Garda** (Northern Trunk): A deep, narrow trench carved by glacial activity, surrounded by steep orography. It reaches the lake's maximum depth of **346 meters** near **Riva del Garda**.
 - **Basso Garda** (Southern Basin): A wider, shallower, semi-circular basin formed by morainic deposits, with depths generally under **80 meters** and gentle bottom gradients.

- **Hydrodynamic Relevance:** This duality allows for simultaneous testing of "deep water" assumptions in the north and "transitional/shallow" physics in the south. The lake is frequently subjected to strong, channeled winds like the *Peler* (northerly) and *Ora* (southerly), or occasional *Foehn* storm events, which can generate significant wave heights over its **52 km** maximum fetch.

This example compares a high-wind scenario against the model's finite-depth physics engine, specifically targeting the shallower southern sector where depth-induced wave breaking and friction become governing factors.

7.1 Scenario Definition

Lake Garda is a deep, elongated lake, but this simulation focuses on a wave propagating into a depth-limited nearshore zone.

- **Meteorological Forcing:** A strong storm event from the South-Southwest (SSW).
- **Wind Speed (U_{10}):** 25.0 m/s (approx. 48 knots).
- **Fetch Length (F):** 45.0 km (Effective straight-line distance).
- **Water Depth (d):** 10.0 m (Simulating a shallow bay or nearshore approach).
- **Duration (t_act):** Assumed Infinite (Steady-state / Fully Developed).

7.2 Step-by-Step Calculation Workflow

Step 1: Atmospheric Adjustment The model first converts the standard meteorological wind speed into the Adjusted Wind Speed (U_A) to match the calibration of the SMB curves.

- **Input:** $U_{10} = 25.0 \text{ m/s}$
- **Calculation:** $U_A = 0.71 \cdot (25.0)^{1.23}$
- **Result:** $U_A = 37.22 \text{ m/s}$
- **Note:** *This significantly increases the effective stress factor used in the wave growth equations.*

Step 2: Limiting Condition Check The software calculates the minimum duration required to generate fetch-limited waves for this distance.

- **Min Duration Required:** 3.24 hours
- **Given Duration:** Infinite
- **Conclusion:** The sea state is **FETCH-LIMITED**. The waves have reached the maximum size possible for a 45 km distance; blowing the wind longer will not increase their size.

7.3 Simulation Results

The physics engine calculates the wave properties using the Hurdle & Stive finite-depth equations, solving the transcendental dispersion relation for wavelength.

Parameter	Value	Unit	Description
Sig. Wave Height (H_s)	2.85	m	Average of highest 1/3rd of waves
Peak Wave Period (T_s)	7.12	s	Time interval between crests
Wavelength (L)	61.18	m	Distance between crests
Wave Celerity (C)	8.59	m/s	Speed of the wave form

7.4 Physical Analysis & Stability

Flow Regime

- **Ratio (d/L):** $10.0/61.18 \approx 0.16$
- **Classification: Transitional / Intermediate Water**
- *Analysis:* The waves are "feeling the bottom." Deep water equations would be inaccurate here. The 2.85 m height is lower than standard deep-water predictions for these conditions (typically ≈ 3.5 m), reflecting the energy dissipation caused by the 10m depth limit.

Miche Stability Criterion (Breaking) The software checks if the waves are too steep to exist without breaking.

- **Calculated Steepness (H/L):** 0.0466
- **Miche Limit (H/L):** 0.1097 (Calculated via $0.142 \cdot \tanh(kd)$)
- **Status: STABLE** (Stability Margin: 57.5%)
- *Conclusion:* While high, these waves are physically sustainable and are not yet at the breaking point induced by depth or steepness.

8. Bibliography

This section cites the foundational scientific documents and revisions for this document.

1. **U.S. Army Corps of Engineers (2008).** *Coastal Engineering Manual (CEM)*. Engineer Manual 1110-2-1100, Washington, D.C.
2. **U.S. Army Corps of Engineers (1984).** *Shore Protection Manual (SPM)*. Volume 1, Coastal Engineering Research Center, Vicksburg, MS.
3. **World Meteorological Organization (2018).** *Guide to Wave Analysis and Forecasting*. WMO-No. 702.
4. **Sverdrup, H.U., and Munk, W.H. (1947).** *Wind, Sea, and Swell: Theory of Relations for Forecasting*. H.O. Pub. No. 601, U.S. Navy Hydrographic Office.
5. **Bretschneider, C.L. (1952).** "Revised Wave Forecasting Relationships." *Proceedings of the 2nd Conference on Coastal Engineering*, ASCE.

6. **Bretschneider, C.L. (1958).** "Revisions in Wave Forecasting: Deep and Shallow Water." *Proceedings of the 6th Conference on Coastal Engineering*, ASCE.
7. **Hurdle, D.P., and Stive, R.J.H. (1989).** "Revision of SPM 1984 Wave Hindcast Model to Avoid Inconsistencies in Engineering Applications." *Coastal Engineering*, 12, 339-351.
8. **Bishop, C.T., Donelan, M.A., and Kahma, K.K. (1992).** "Shore Protection Manual's Wave Prediction Reviewed." *Coastal Engineering*, 17, 25-48.
9. **Etemad-Shahidi, A., Kazeminezhad, M.H., and Mousavi, S.J. (2009).** "On the Prediction of Wave Parameters Using Simplified Methods." *Journal of Coastal Research*, SI 56, 505-509.
10. **Miche, M. (1944).** "Mouvements ondulatoires de la mer en profondeur constante ou décroissante." *Annales des Ponts et Chaussées*.
11. **Fenton, J.D. (1999).** "Numerical methods for nonlinear waves." In P.L.-F. Liu (Ed.), *Advances in Coastal and Ocean Engineering* (Vol. 5, pp. 241–324). World Scientific: Singapore.