OFFSHORE-TO-NEARSHORE WAVE TRANSFORMATION

(Simplified Approach Using Linear Wave Theory)

Overview

This program processes wave data from an input CSV file, computes nearshore wave parameters at a specified depth, and generates:

- **`output.csv`** – Contains the computed results.
- **`report.txt`** – Provides descriptive statistics of both input and computed variables.

USAGE

```
./transpose input_csv coast_dir depth_d
```

Arguments:

- **`input_csv`** : CSV input file (with columns: `datetime`, `swh`, `mwd`, `pp1d`)
- **`coast_dir`** : Coastline orientation in degrees (clockwise from North)
- **`depth_d`** : Local depth (meters)

CSV INPUT FORMAT

The input CSV file should be comma-separated with at least the following columns:

```
datetime, swh, mwd, pp1d, [additional columns ignored]
```

OUTPUT CSV FORMAT

The generated `output.csv` will contain the following comma-separated columns:

```
datetime,swh_offshore,mwd_offshore,pp1d,L0,depth_d,L,kh,alpha_offshore,a
```

Computed Parameters

Parameter	Description
L0	Deep-water wavelength: `L0 = g * T² / (2π)`
depth_d	Local depth (input parameter echoed in output)

Parameter	Description
L	Local wavelength, solved from `L = L0 * tanh((2π * depth_d) / L)`
kh	Wave number (`k = 2π / L`) times local depth (`h`)
alpha_offshore	Offshore wave approach angle relative to coastline
alpha_local	Local wave angle after refraction
mwd_local	Local mean wave direction, adjusted from offshore `mwd`
Ks	Shoaling coefficient
Kr	Refraction coefficient
Hb	Breaking wave height (Miche, 1944): `Hb = 0.142 * L * tanh((2π * depth_d) / L)`
swh_local	Local significant wave height (minimum of `swh * Ks * Kr` and `Hb`)

Note: Waves arriving from directions between `coast_dir` and `coast_dir + 180°` (i.e., from the land side) are set to **zero**.

COMPILATION

To compile the program, use the following command:

```
g++ -O3 -fopenmp -march=native -std=c++17 -Wall -Wextra -pedantic -Wconversion -Wsign-conversion -static -static-libgcc -static-libstdc++ -o transpose transpose.cpp
```

This command enables **optimizations** and includes several **compiler warnings** to ensure code quality.
