Resonant triad interactions of gravity waves in cylindrical basins
Supplementary Material

Matthew Durey & Paul A. Milewski

We here outline the structure of the MATLAB code used to generate resonant triads of gravity waves in a circular cylinder of finite depth (corresponding to Table 1 presented in the main text). The main feature of the code is the application of Theorem 1 to assess whether a triad exists for three given wave modes satisfying the correlation condition; when the triad exists, we compute the corresponding critical depth. Questions regarding the implementation of the code should be addressed to Matthew Durey (matthew.durey@glasgow.ac.uk). The code was written and tested using MATLAB 2018a.

We consider a triad whose angular frequencies satisfy $\Omega_1 + \Omega_2 = \Omega_3$ and azimuthal wavenumbers satisfy $m_1 + m_2 = m_3$ (or `om1` + `om2` = `om3` and `m1` + `m2` = `m3` in the code). The script

`Circular_cylinder_find_triads.m`

exhaustively loops through all possible combinations of three wave modes, checking whether the triad existence condition (Theorem 1) is satisfied for each combination. To reduce the number of combinations, we introduce the positive integers `m_max` and `n_max`, which bound the maximum allowable values of the azimuthal wavenumbers (`m1`, `m2`, `m3`) and radial wavenumber indices (`n1`, `n2`, `n3`) (lines 34 and 35). To reproduce the results in Table 1 of the main text, the user should set `m_max` = 3 and `n_max` = 3; however, different values (and thus more triad combinations) may be considered by the user. To further reduce the number of combinations considered in the `for` loop (lines 47 to 115), we consider the following restrictions

- $m_3 \geq 0$, where the case $m_3 < 0$ may be recovered by mapping $(m_1, m_2) \mapsto (-m_1, -m_2)$.
- $m_1 \leq m_2$, where the case $m_1 > m_2$ may be recovered by swapping the roles of modes 1 and 2.

The main computations are summarised as follows. In lines 37 to 39, we define the angular frequency for a given wavenumber and fluid depth. In lines 40 to 45, we define storage vectors, which will be populated by elements characterising each triad, namely the radial wavenumbers (`k1_vec`, etc.), azimuthal wavenumbers (`m1_vec`, etc.), radial wavenumber indices (`n1_vec`, etc.), and the critical depths (`h_vec`). The main computations are performed in lines 46 to 115, where we loop over all permissible values of m_1, m_2 and m_3 (satisfying $m_1 + m_2 = m_3$), and then proceed as follows:

1. Determine all the admissible radial wavenumbers (K_1, K_2 and K_3, denoted `k1`, `k2` and `k3` in the code), with each radial wavenumber satisfying the radial no-flux boundary condition (lines 52 to 75). This calculation uses the `zerobess.m` function developed by Jonas Lundgren for calculating the roots of Bessel functions of first kind (downloaded from MATLAB Central).
2. Loop through the candidate triads for given `k1`, `k2` and `k3` (lines 76 to 112).
3. Determine if the resonance condition (see Theorem 1) is satisfied for the given combination of wave modes (lines 86 to 88). If the resonance condition is not satisfied, then the code skips steps 4 and 5 below, and moves to the next mode combination.
4. If the mode combination gives rise to a resonant triad, then compute the critical depth, `h` (lines 89 to 94).

5. Store the values characterising the triad, appending to the existing storage vectors.

Finally, we compute the mean radial wavenumber, $\overline{K} = \frac{1}{3}(K_1 + K_2 + K_3)$, for each triad (denoted `kmean`) in lines 116 to 118. Notably, some values of the critical depth may be repeated; this scenario corresponds to the same triad being identified, but with the roles of modes 1 and 2 swapped.

We compute the correlation integral, $\mathscr{C}$ (defined analogously to equation (4.7) in the main text, but with Ψ_3 replaced by Ψ_3^*), in lines 119 to 140. The code first defines vector functions corresponding to the unnormalised Bessel functions defining each mode in the triad (`B1_vec`, etc.). The normalisation factors (`N1_vec`, etc.) are then computed (lines 125 to 129), and coupled with the Bessel functions to determine the normalised radial functions `C1`, `C2` and `C3` (lines 130 to 134). We then compute the correlation integral, with the radial parts computed using the MATLAB function `integral` and the angular parts computed analytically (giving 2π). Notably, the area, S, of the cylinder cross section is equal to π, whose reciprocal appears as a prefactor in the definition of $\mathscr{C}$; as such, the prefactor of `2` in line 139 arises from simplifying $2\pi/\pi$, with the 2π being the angular integral and π being the cross-sectional area. When the values of `m_max` and `n_max` are large, line 139 is very slow to compute; the user may prefer to exclude this line (and subsequent uses of `C_int`) from the computation in such an eventuality.

The remainder of the code summarises the triads in various ways, and the user may choose to exclude any of these sections. We briefly outline the different sections as follows.

- Define the β coefficient (see equation (4.11) in the main text) for each triad, as appears in the weakly nonlinear analysis (lines 141 to 151). Here we map $\Omega_3 \mapsto -\Omega_3$ so that $\Omega_1 + \Omega_2 + \Omega_3 = 0$, consistent with our framework for the multiple-scales analysis.

- Plot the value of K_1/K_3 and K_2/K_3 for each triad, with each data point colour-coded by the normalised critical depth, $K_3 h_c$. The black curves denote the bounds of the triad existence region (see Theorem 1).

- Plot all values of the critical depth, from largest to smallest (lines 175 to 189). The user may also wish to plot all the unique values of the critical depth, changing line 180 to

$$[\sim, \mathtt{h\_ind}] = \mathtt{sort}(\mathtt{unique}(\mathtt{h\_vec}), '\mathtt{descend}');$$

- Plot the histogram of critical depths, with a logarithmic (base 10) horizontal scale (lines 190 to 198). When large values of `m_max` and `n_max` are considered, this plot highlights the large number of triads that exist only in shallower fluids.

- Plot the quantity $\mathscr{C}\beta$ (lines 199 to 211), as is used to define the coefficients, α_j, arising in the weakly nonlinear analysis (equation 4.10). For this section of code to run, lines 141 to 151 must be run first (so as to define β).

- Output a table similar to Table 1 in the main text (lines 212 to 253), with the triads ordered by decreasing critical depth. The user may choose whether to show repeated critical depths or not (corresponding to the same triad but with modes 1 and 2 swapped): to show all triads, including depth repetitions, set `show_repeated_triads` = `true` in line 219; to show only unique values of the critical depth, set `show_repeated_triads` = `false`. The user may also wish to choose the maximum number of triads to display in the table, defining the positive integer `max_number_to_display` in line 238. If the number of triads is smaller than `max_number_to_display`, then all triads are listed. Below the table, we output statements summarising whether repeated triads are included in the table or not, and how many triads are presented in the table (lines 244 to 253).