Manual for

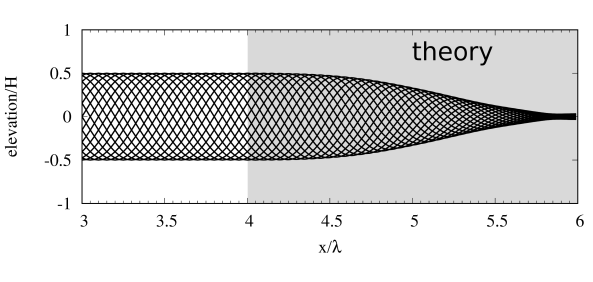
CRest.py

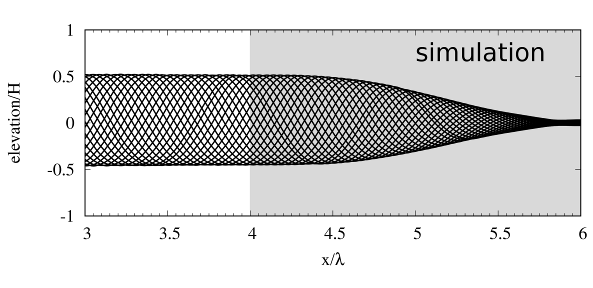
a computer program for
estimating reflection coefficient C_R for forcing zones
(wave absorbing layers, sponge layers, damping zones, etc.)
in flow simulations of free-surface wave propagation

written by Robinson Perić

at the Institute for Fluid Dynamics and Ship Theory,
Hamburg University of Technology (TUHH), Germany

version August 12, 2019





Contents

1 One-minute-explanation of the code

When to use this code?

- You perform flow simulations with free-surface wave propagation in 2D or 3D
- You want to minimize undesired wave reflections at the domain boundaries
- To do so, you want to use *forcing-zone-type approaches* (e.g. wave absorbing layers, sponge layers, damping zones, Euler overlay method, relaxation zones[1], etc.)

What does the code do?

- The user can enter wave period, water depth, wavelength, zone thickness, blending function, and specify in which governing equations the source terms are introduced
- The code calculates reflection coefficient C_{R} for various forcing strengths γ according to [1] and writes these results to the file ′C__R.csv′ in the same folder
- If matplotlib is installed: A window will open with an interactive plot of C_{R} over γ
- Thus the code can be used to quickly fine-tune the case-dependent forcing zone parameters

Requirements:

- Check that your forcing zone can be formulated in terms of Eqs. (5) and (4)
- Install *python* programming language
- For full functionality, install *matplotlib*
- Operating systems: Linux, macOS, Windows

Tuning forcing zones:

- Use blending functions such as Eqs. (8) to (13)
- Increasing the zone thickness x_{d} tends to lower the reflection coefficient and to widen the range of wavelengths which will be damped satisfactorily
- Typically, confidence in minimizing reflections is more important than the last few percent efficiency $\Rightarrow$ use slightly thicker zones than necessary
- Common values for zone thickness are $1\lambda \leq x_{\mathrm{d}} \leq 2\lambda$
- For irregular waves, tune the forcing zone to the peak period or the longest period (quick approach [2]) or calculate the reflection coefficient for each wave component's period and then tune accordingly (more accurate approach)

Benefits and Limitations:

- The code satisfactorily predicted reflection coefficients and optimum forcing zone parameters in 2D- and 3D-flow simulations with regular, irregular, linear, and highly nonlinear waves in shallow, intermediate and deep water depth
- For complex flows (e.g. steep waves, breaking waves or 3D oblique wave incidence), reflection coefficients were in some cases slightly larger than predicted, so that thicker zones are

[1]Relaxation zones are a special case of forcing zones, which simultaneously blend-out the governing equations and blend-in a reference solution. To tune relaxation zones, please use the following code: `https://github.com/wave-absorbing-layers/relaxation-zones-for-free-surface-waves`

recommended to ensure satisfactory wave absorption

- Undesired reflections can also be due other mechanisms, e.g. the use of inappropriate grids or certain forcing zone arrangements
- Therefore, tuning the forcing zone parameters according to this code does NOT guarantee that the actual reflection coefficient in the simulation will equal the prediction

2 Requirements

The programming language python version 2.7 or 3.0 or higher (`https://www.python.org/downloads/`) must be installed.

It is recommended to also have matplotlib (`https://matplotlib.org/users/installing.html`) installed[2]. Then the code will open a window with an interactive plot of the results.

3 Motivation and theory behind the code

Forcing zones[3] (wave absorbing layers, sponge layers, damping zones, numerical beaches, ...) reduce undesired wave reflections at boundaries of the computational domain by introducing source terms in the governing equations to force the flow towards a prescribed reference solution. However, the forcing function contains three user-defined parameters: forcing strength γ, blending function $b(\mathbf{x})$, and zone thickness x_d; these parameters are case-dependent and must be tuned for every simulation. Otherwise strong reflection may occur [2].

The present code can guide the tuning of these case-dependent parameters. The code is an implementation of the theory from [1], where an analytical approach was presented which predicts the reflection coefficient for a given forcing zone setup. The reflection coefficient C_R is the ratio of the reflected wave amplitude to the incidence wave amplitude. Thus the theory can be used to tune the case-dependent parameters *before running the simulation.*

For practical discretizations, forcing zones were found to behave independent of the discretization and the used flow solver, so the present code applies to most implementations of forcing-zone-type approaches.

The code's predictions for reflection coefficient and optimum forcing zone parameters were demonstrated to be of satisfactory accuracy in 2D- and 3D-flow simulations with regular, irregular, linear, and highly nonlinear waves in shallow, intermediate and deep water depths (cf. [1]–[7]). At first glance, this may seem surprising, since the code considers 1D-wave-propagation based on linear wave theory; why the code is also recommended for tuning forcing zones in 3D-flow simulations with strongly reflecting bodies subjected to free-surface waves is outlined in the following.

[2]For debain-based linux systems, install e.g. via: `sudo apt-get install python3-matplotlib`.

[3]Some names, such as 'absorbing layers' or 'damping zones', are to some extent misleading: for example, for absorption or damping phenomena one would expect that when more absorption or damping is introduced, then there should be less wave reflection. In contrast, reflection actually increases when the source terms become larger than optimum. This can be explained by looking at a single cell in which the velocity is forced towards zero with an infinitely large source term: then the velocity in the cell will be zero, and there will be no flow in or out of the cell; thus the cell will behave like a wall, so waves will be fully reflected at it. Thus the term 'forcing zones' is used in the following, since it more accurately describes what these approaches actually do.

Forcing zones reduce undesired wave reflections via three mechanisms: wave absorption, partial wave reflection occurring everywhere within the forcing zone where the gradient of the blending function $\nabla b(\mathbf{x}) \neq 0$, and destructive interference of these partial wave reflections. All three mechanisms are considered in the present code.

When a forcing zone is optimally tuned, then the wave height gradually decreases within the forcing zone (cf. plots on title page with forcing zone shaded in gray) and partial wave reflections occur throughout the whole forcing zone with small amplitude; thus when optimally tuned, forcing zones 'split up' nonlinear waves into many approximately linear wave components, which interfere destructively due to their phase differences [7].

This behavior characterizes a key difference between forcing zone approaches and boundary-based wave absorption techniques such as absorbing boundary conditions or active wave absorption: Boundary-based approaches can neither utilize partial wave reflections nor their destructive interference, since they act only at the boundary. This explains why forcing zones, despite their simple formulation, are capable of damping even highly nonlinear waves in complex flow problems, where high-order boundary-based wave damping approaches would fail.

In 3D-flows, when waves are reflected at a body within the domain, some of these waves will be re-reflected twice at the domain boundaries (i.e. they encounter two forcing zones and thus are 'damped twice') before they travel back towards the domain center. This was found to improve the wave absorption, so overall reflection coefficients in 3D-flow simulations are mostly lower than in the 2D-case. Extension of the theory towards oblique wave incidence demonstrated that already the theory for 1D-wave-propagation (i.e. long-crested waves traveling in one direction, i.e. 2D-flow) provides a decent estimate of the overall reflection coefficient for typical 3D-flow simulations, where a wave-reflecting body is positioned in the domain center [7].

Thus the present code can be recommended for practical 3D-flow simulations featuring nonlinear wave phenomena.

Future research will provide further details to more accurately assess the accuracy of the theory's predictions. In the meantime, it is recommended to select the forcing zone thickness slightly larger than theoretically necessary, to ensure satisfactory wave absorption.

The theory in [1] was derived so that it works for any continuous or discontinuous blending function. A few common blending functions are already implemented. Custom blending functions can be entered at the location indicated in the source code.

4 Introduction to forcing zones and other wave absorbing layers

The theory in [1] is based on the following forcing zone formulation, which applies to many existing forcing zone-type approaches. Consider e.g. the following governing equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho \,\mathrm{d}V + \int_S \rho(\mathbf{v} - \mathbf{v}_g) \cdot \mathbf{n} \,\mathrm{d}S = 0 \quad , \tag{1}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho u_i \, \mathrm{d}V + \int_S \rho u_i(\mathbf{v}-\mathbf{v}_g)\cdot\mathbf{n}\,\mathrm{d}S = \int_S (\tau_{ij}\mathbf{i}_j - p\mathbf{i}_i)\cdot\mathbf{n}\,\mathrm{d}S + \int_V \rho\mathbf{g}\cdot\mathbf{i}_i\,\mathrm{d}V + \int_V \rho q_i\,\mathrm{d}V \quad , \tag{2}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \alpha\,\mathrm{d}V + \int_S \alpha(\mathbf{v}-\mathbf{v}_\mathrm{g})\cdot\mathbf{n}\,\mathrm{d}S = \int_V q_\alpha\,\mathrm{d}V \quad . \tag{3}$$

Here V is the control volume (CV) bounded by the closed surface S, $\mathbf{v}$ is the velocity vector of the fluid with the Cartesian components u_i, $\mathbf{v}_g$ is the grid velocity, $\mathbf{n}$ is the unit vector normal to S and pointing outwards, t is time, p is the pressure, ρ are fluid density, τ_{ij} are the components of the viscous stress tensor, $\mathbf{i}_j$ is the unit vector in direction x_j, with volume fraction α of water.

Then undesired wave reflections can be reduced by applying source terms for volume fraction, q_α, and momentum, q_i, as

$$q_\alpha = \gamma b(\mathbf{x})\left(\alpha_\mathrm{ref} - \alpha\right) \quad , \tag{4}$$

$$q_i = \gamma b(\mathbf{x})(u_{i,\mathrm{ref}} - u_i) \quad , \tag{5}$$

with reference volume fraction α_ref, reference velocity component $u_{i,\mathrm{ref}}$, forcing strength γ and blending function $b(\mathbf{x})$; see [1] for details. Outside the forcing zone holds $q_i = q_\alpha = 0$.

The forcing strength γ has the unit $\left[\frac{1}{\mathrm{s}}\right]$ and regulates how strong the solution at a given cell is forced towards the reference solution.

Blending function $b(\mathbf{x})$ regulates how the magnitude of the source term varies within the forcing zone. Outside the forcing zone holds $b(\mathbf{x}) = 0$, and inside the forcing zone holds $0 \leq b(\mathbf{x}) \leq 1$. Many different types of blending functions can be applied. The following common choices are implemented in the code:

Constant blending

$$b(\mathbf{x}) = 1 \quad , \tag{6}$$

linear blending

$$b(\mathbf{x}) = \left(\frac{x_\mathrm{d} - \tilde{x}}{x_\mathrm{d}}\right) \quad , \tag{7}$$

quadratic blending

$$b(\mathbf{x}) = \left(\frac{x_\mathrm{d} - \tilde{x}}{x_\mathrm{d}}\right)^2 \quad , \tag{8}$$

$\cos^2$-blending

$$b(\mathbf{x}) = \cos^2\left(\frac{\pi}{2} + \frac{\pi}{2}\left(\frac{x_\mathrm{d} - \tilde{x}}{x_\mathrm{d}}\right)\right) \quad , \tag{9}$$

exponential blending

$$b(\mathbf{x}) = \left(\frac{e^{((x_\mathrm{d}-\tilde{x})/x_\mathrm{d})^2} - 1}{e^1 - 1}\right) \quad , \tag{10}$$

power blending

$$b(\mathbf{x}) = \left(\frac{x_\mathrm{d} - \tilde{x}}{x_\mathrm{d}}\right)^n \quad , \tag{11}$$

exponential blending with power n

$$b(\mathbf{x}) = \left(\frac{e^{((x_\mathrm{d}-\tilde{x})/x_\mathrm{d})^n} - 1}{e^1 - 1}\right) \quad , \tag{12}$$

and $\cos^{2n}$-blending

$$b(\mathbf{x}) = \left[\cos^2\left(\frac{\pi}{2} + \frac{\pi}{2}\left(\frac{x_\mathrm{d} - \tilde{x}}{x_\mathrm{d}}\right)\right)\right]^n \quad , \tag{13}$$

where $\tilde{x}$ is the shortest distance of location $\mathbf{x}$ to the closest domain boundary to which a forcing zone of thickness x_d is attached, and n regulates the shape of the blending function. Some of these blending functions are illustrated in Fig. 1.

Though so far the optimum blending function is not known, several investigations showed that typically constant or linear blending functions are not recommended since they are typically less effective.

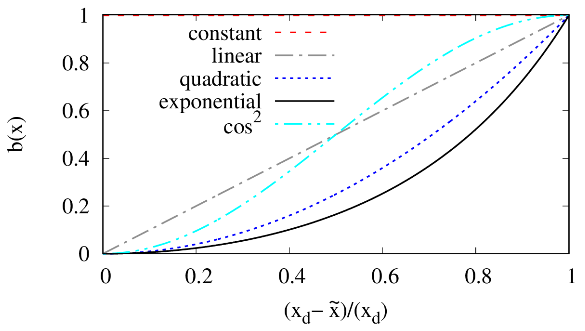


Figure 1: Different blending functions $b(\mathbf{x})$ over $\mathbf{x}$-location in the forcing zone; $\tilde{x}$ is the shortest distance of $\mathbf{x}$ to the closest domain boundary to which a forcing zone of thickness x_d is attached; waves enter the zone at $(x_\mathrm{d} - \tilde{x})/x_\mathrm{d} = 0$, and the boundary to which the zone is attached lies at $(x_\mathrm{d} - \tilde{x})/x_\mathrm{d} = 1$; for constant (Eq. (6)), linear (Eq. (7)), quadratic (Eq. (8)), exponential (Eq. (10)), and $\cos^2$ (Eq. (9)) blending

As shown in [2], parameters γ and x_d scale with angular wave frequency ω and wavelength λ as

$$\gamma \propto \omega \quad , \quad x_\mathrm{d} \propto \lambda \quad . \tag{14}$$

Therefore these parameters must be tuned for every simulation and default coefficients should not be used.

Forcing may be applied only for horizontal momentum (q_x, q_y), or for vertical momentum (q_z), or for volume fraction (q_α), or for any combination of these. If the reference solution is the hydrostatic solution for the undisturbed free surface (e.g. $u_{i,\mathrm{ref}} = 0$), then the forcing can be interpreted as '*wave damping*'.

Further, different 'forcing zone arrangements' may be used, i.e. forcing zones may be attached to one or several domain boundaries, source terms can be applied in different governing equations and the flow can be forced towards different reference solutions as illustrated in Fig. 2.

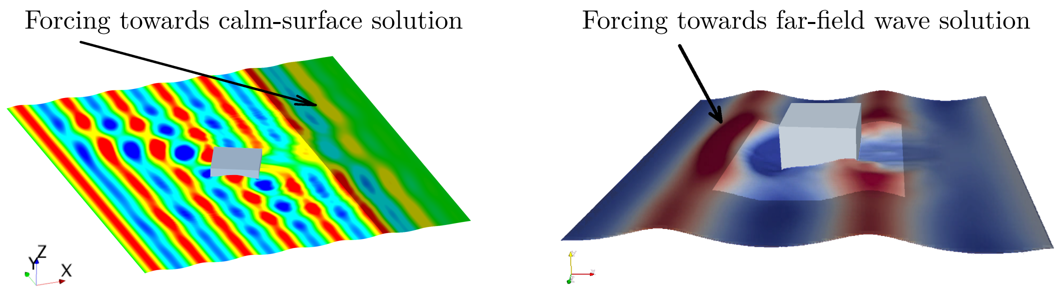

Forcing towards calm-surface solution
Forcing towards far-field wave solution
Y
Z
X

Figure 2: Examples of different forcing zone arrangements used in practice; the color variations denote the surface elevation; for a floating pontoon (gray box) subjected to surface waves; with forcing zones (shaded gray); forcing velocities towards zero in the vicinity of the domain outlet (left) or forcing of volume fraction and all velocities towards the far-field wave solution near all vertical domain boundaries

IMPORTANT

If your governing equations look more like Eq. (15) instead of Eqs. (1–3), then your implementation likely is a special case of forcing zones, often called 'relaxation zones'. **If so, do not use the present code!** Instead, please use the following code: `https://github.com/wave-absorbing-layers/relaxation-zones-for-free-surface-waves`

The link also contains a manual with a more detailed discussion of the following brief overview: Relaxation zones blend, say a general transport equation $\mathcal{T}$ for transport quantity ϕ, over to a reference solution via

$$(1 - b(\tilde{x}))\,\mathcal{T} + \frac{b(\tilde{x})}{\tau}\mathcal{R} = 0 \quad , \tag{15}$$

where $b(\tilde{x})$ is a blending function such as e.g. Eqs. (7) to (13), $\mathcal{T}$ corresponds e.g. to the conservation equations for mass or momentum (without relaxation source terms), and $\mathcal{R}$ corresponds to $\int_V (\phi - \phi_{\text{ref}})\ \mathrm{d}V$ with reference solution ϕ_{ref} for transport quantity ϕ. The relaxation parameter τ has unit [s] and regulates the magnitude of the source term in such a way that a large value of τ implicates a small source term and vice versa; note that sometimes in literature Eq. (15) is written without τ, which is explained under the above link.

5 Benefits and limitations

The code's predictions for reflection coefficient and optimum forcing zone parameters were demonstrated to be of satisfactory accuracy in 2D- and 3D-flow simulations with regular, irregular, linear, and highly nonlinear waves (cf. [1]–[7]). Reflection coefficients in flow simulations were typically lower or similar to the code's predictions, but never more than a few percent larger when optimally tuned using the code.

Although the code's predictions are usually quite accurate [1], please keep in mind that every theory has its limitations! For highly non-linear waves, such as breaking waves, or for complex 3D-flows with oblique wave incidence, reflection may be larger than predicted. Further, undesired reflections can be due other mechanisms as well, e.g. the use of inappropriate grids or certain forcing zone arrangements[4]. Therefore, tuning the forcing zone's parameters according to the present theory

[4]For example, forcing fluid momentum and volume fraction towards the far-field wave solution can create flow disturbances when the forcing zones are tangential to the wave propagation direction, if there is a mismatch between the computed flow solution within the domain (including discretization and iteration errors) and the reference solution (without discretization and iteration errors). Such flow disturbances can be radiated as undesired waves into the

does NOT guarantee that the actual reflection coefficient in the simulation will equal the prediction. See [1] and [7] for a detailed discussion.

6 Recommendations

The use of blending functions such as Eqs. (8) to (12) is recommended. Constant or linear blending, i.e. Eqs. (6) and (7), are typically less efficient, meaning that to obtain the same reflection coefficient they require greater zone thickness and thus also greater computational effort. Currently it is not clear which blending function is the best choice: Although blending functions in Eqs. (8) to (10) look different, the differences in wave absorption between them was comparatively small, with perhaps a slight preference towards exponential blending [1].

Increasing the zone thickness x_d tends to widen the range of wavelengths which will be damped satisfactorily and to lower the reflection coefficient at the optimum parameter setting; thus if the wave absorption is not satisfactory, then zone thickness x_d should be increased.

Although for regular waves it may be possible to achieve satisfactory damping with zones as thin as $x_\mathrm{d} \approx 0.5\lambda$, such thin zones should be avoided or at least used with caution, since then the reflection coefficient can be very sensitive to the wave parameters.

In engineering practice, usually confidence in wave absorption is more important than the last few percent efficiency. Therefore it is recommended to use zone thicknesses x_d slightly larger than possibly necessary. Typical values are $1\lambda \leq x_\mathrm{d} \leq 2\lambda$, in terms of wavelength λ.

For irregular waves, a quick approach is to tune the zone to the peak period or the longest period[2, 7]. A more accurate approach is to calculate the reflection coefficient for each wave component of the wave energy spectrum and in this manner tune the zone parameters accordingly.

7 How to run the code

In windows, double click on the executable file *CRest.py.*

In Linux and macOS, open a terminal and type:

```
python CRest.py
```

Example output of the code is shown in Fig. 3.

Alternatively, experienced users may pass the parameters as arguments[5].

domain. If the reference solution is highly accurate, these disturbances vanish on fine grids; however, finer grids than commonly used may be required. Luckily, there are several forcing zone arrangements that do not have these problems, see [7].

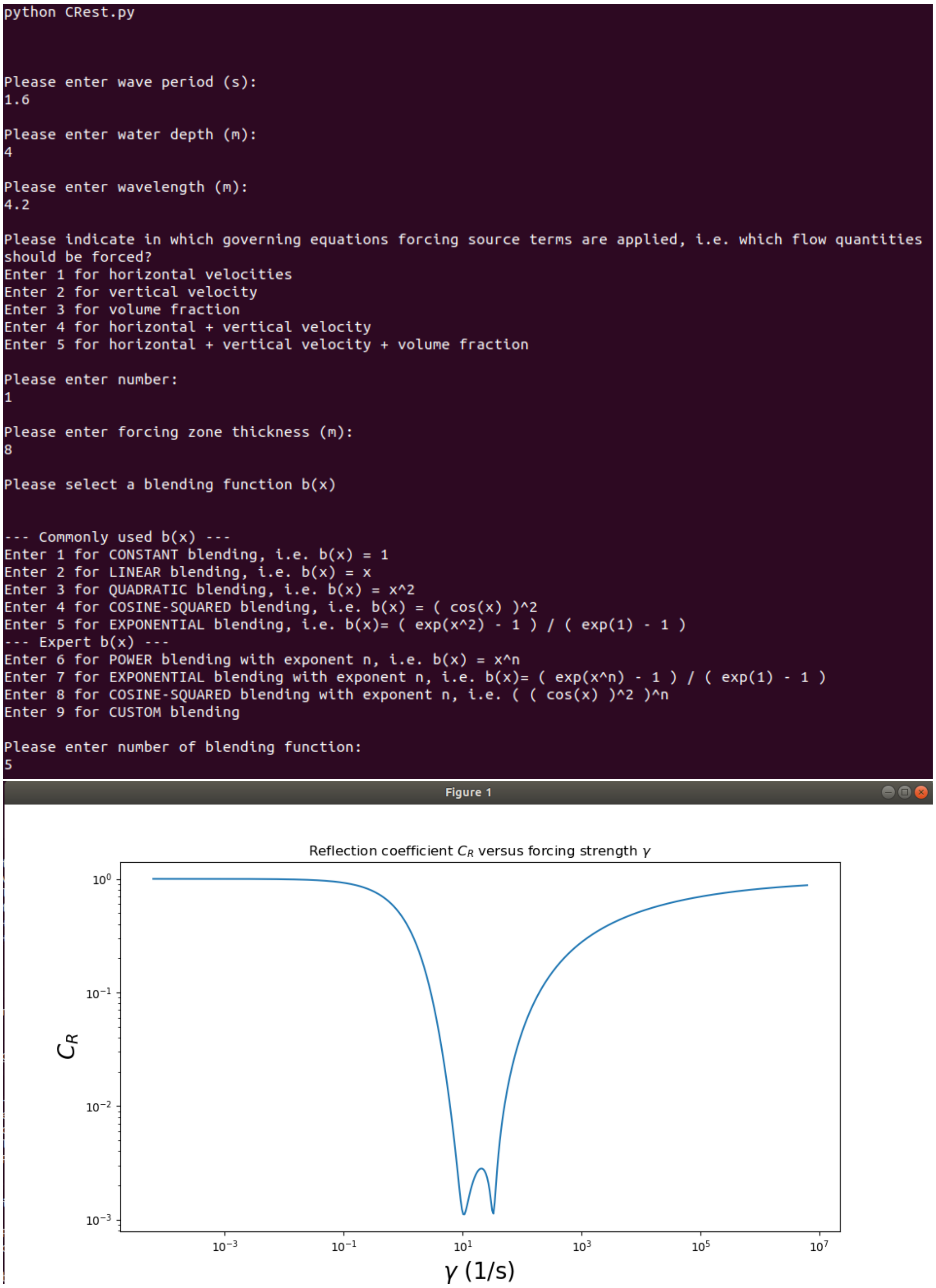
[5]To use the code in batch-mode for blending via Eqs. (6-10), type:

```
python CRest.py T h L forcedEQs xd b
```

and replace `T` with the wave period in seconds, `h` with the water depth in meters, `L` with the wavelength in meters, `forcedEQs` by the number representing which flow quantities forcing is applied to (cf. Fig. 3), `xd` by the zone thickness in meters, and `b` by the number representing the desired blending function (cf. Fig. 3).
To use the code in batch-mode for blending via Eqs. (11-13), type:

```
python CRest.py T h L forcedEQs xd b n
```

and replace n by the blending function exponent



```
python CRest.py


Please enter wave period (s):
1.6

Please enter water depth (m):
4

Please enter wavelength (m):
4.2

Please indicate in which governing equations forcing source terms are applied, i.e. which flow quantities
should be forced?
Enter 1 for horizontal velocities
Enter 2 for vertical velocity
Enter 3 for volume fraction
Enter 4 for horizontal + vertical velocity
Enter 5 for horizontal + vertical velocity + volume fraction

Please enter number:
1

Please enter forcing zone thickness (m):
8

Please select a blending function b(x)


--- Commonly used b(x) ---
Enter 1 for CONSTANT blending, i.e. b(x) = 1
Enter 2 for LINEAR blending, i.e. b(x) = x
Enter 3 for QUADRATIC blending, i.e. b(x) = x^2
Enter 4 for COSINE-SQUARED blending, i.e. b(x) = ( cos(x) )^2
Enter 5 for EXPONENTIAL blending, i.e. b(x)= ( exp(x^2) - 1 ) / ( exp(1) - 1 )
--- Expert b(x) ---
Enter 6 for POWER blending with exponent n, i.e. b(x) = x^n
Enter 7 for EXPONENTIAL blending with exponent n, i.e. b(x)= ( exp(x^n) - 1 ) / ( exp(1) - 1 )
Enter 8 for COSINE-SQUARED blending with exponent n, i.e. ( ( cos(x) )^2 )^n
Enter 9 for CUSTOM blending

Please enter number of blending function:
5
```

Figure 3: Example output of the code to compute the reflection coefficient C_R depending on the value of forcing strength γ; for a wave with period $T = 1.6\,\mathrm{s}$, water depth $h = 4\,\mathrm{m}$, wavelength $\lambda = 4.2\,\mathrm{m}$; for applying forcing of the horizontal velocities with a forcing zone of thickness $x_\mathrm{d} = 8\,\mathrm{m}$ and exponential blending via Eq. (10)

8 Reporting bugs

Currently, no bugs are known.

If you find bugs, or if you have questions or suggestions, please contact the author:

Robinson Perić
Hamburg University of Technology (TUHH)
Institute for Fluid Dynamics and Ship Theory (M8)
Am Schwarzenberg-Campus 4
D-21073 Hamburg, Germany
Room C5.004
Phone: +49 40 42878 6031
Fax: +49 40 42878 6055
E-mail: robinson.peric@tuhh.de
URL: http://www.tuhh.de/fds/staff/

The code is available at
`https://github.com/wave-absorbing-layers/absorbing-layer-for-free-surface-waves`
Updates and further useful information will be posted there as well.

9 Bug fixes & previous versions of the code

The 2017 version of this code produced inaccurate predictions for forcing of the vertical velocities in shallow water. This has been amended, and now the code should work for all water depths and all forcing zone settings. Furthermore, the interactive-mode was adjusted to be more user-friendly, and a batch mode was added, so parameters can also be passed as arguments.

10 Copyright

The program is published as free software under the GNU General Public License (GPLv3). It would be warmly appreciated if users would cite the corresponding papers in their publications and mention that they used the present code.

11 References

[1] Perić, R., & Abdel-Maksoud, M. (2018). Analytical prediction of reflection coefficients for wave absorbing layers in flow simulations of regular free-surface waves. Ocean Engineering, 147, 132-147.

[2] Perić, R., & Abdel-Maksoud, M. (2016). Reliable damping of free-surface waves in numerical simulations. Ship Technology Research, 63, 1, 1–13.

[3] Perić M. (2015). Steigerung der Effizienz von maritimen CFD-Simulationen durch Kopplung verschiedener Verfahren, in: Jahrbuch der Schiffbautechnischen Gesellschaft, 109. Band, Schiffahrts-Verlag "Hansa" GmbH & Co. KG, Hamburg, 69–76.

[4] Perić, R., & Abdel-Maksoud, M. (2015). Assessment of uncertainty due to wave reflections in experiments via numerical flow simulations. Proc. Twenty-fifth Int. Ocean and Polar Eng. Conf. (ISOPE2015), Hawaii, USA.

[5] Perić, R., Hoffmann, N., & Chabchoub, A. (2015). Initial wave breaking dynamics of Peregrine-type rogue waves: a numerical and experimental study. European J. Mechanics-B/Fluids, 49, 71–76.

[6] Perić, R., Vukčević, V., Abdel-Maksoud, M., & Jasak, H. (2018). Tuning the Case-Dependent Parameters of Relaxation Zones for Flow Simulations With Strongly Reflecting Bodies in Free-Surface Waves. arXiv preprint arXiv:1806.10995.

[7] Perić, R., & Abdel-Maksoud, M. (2019). Reducing Undesired Wave Reflection at Domain Boundaries in 3D Finite Volume–Based Flow Simulations via Forcing Zones. Journal of Ship Research.