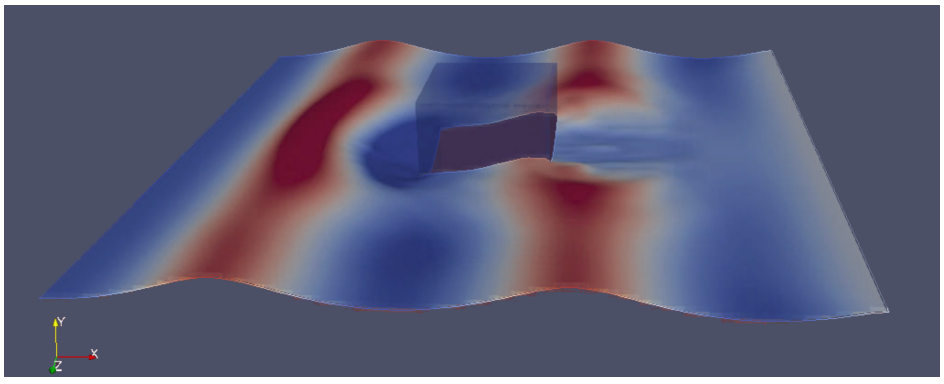
# Manual for *CRestRelax.py*

**a computer program for**
**estimating reflection coefficient $C_{\mathrm{R}}$ for relaxation zones**
**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
and the Fakultet Strojarstva i Brodogradnje,
Sveučilište u Zagrebu

version June 29, 2018



# Contents

# 1 One-minute-explanation of the code

When to use this code?

- You perform flow simulations of free-surface wave propagation
- You want to minimize undesired wave reflections at the domain boundaries
- To do so, you want to use *relaxation zones*, which blend the flow solution towards the far-field wave in a zone usually attached to the vertical domain boundaries

What does the code do?

- The user can enter wave period, wavelength, zone thickness, blending function $b(\tilde{x})$, and blending exponent $n$
- The code calculates reflection coefficient $C_\mathrm{R}$ for various values of relaxation zone parameters $\tau$ according to [1] and writes these results to the file $'\mathrm{C_R}.\mathrm{csv}'$ in the same folder
- If matplotlib is installed: A window will open with an interactive plot of $C_\mathrm{R}$ over $\tau$
- Thus the code can be used to quickly fine-tune the case-dependent relaxation zone parameters

Requirements:

- Check that your relaxation zone can be formulated in terms of Eqs. (2) and (3)
- Install *python* programming language
- For full functionality, install *matplotlib*
- Operating systems: Linux, macOS, Windows

Tuning forcing zones:

- Increasing the zone thickness $x_\mathrm{d}$ widens the range of wavelengths which will be damped satisfactorily, and lowers the reflection coefficient for the optimum setting
- Often confidence in reflection absorption is more important than the last few percent efficiency $\Rightarrow$ use slightly thicker zones than necessary
- Typical values for zones thickness are $1\lambda \leq x_\mathrm{d} \leq 2\lambda$
- For irregular waves, tune the zone to the peak period or the longest period (quick approach [3]) or calculate the reflection coefficient for each wave component's period and then tune accordingly (more accurate approach)

Benefits and Limitations:

- The code gave satisfactory predictions for regular long-crested waves in both shallow and deep water; results for steep waves, irregular waves, shallow water waves, breaking waves and 3D flow problems with oblique wave incidence indicate that the theory gives satisfactory results for such cases as well
- However, for complex flows (e.g. steep waves, breaking waves or 3D oblique wave incidence) reflection may be larger than predicted, so that thicker zones are recommended to ensure satisfactory wave absorption
- Undesired reflections can also be due to other mechanisms, e.g. the use of inappropriate grids

- Therefore, tuning the relaxation 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. Then the code will open a window with an interactive plot of the results.

The code was tested with Ubunut Linux 18.04, and the following packages installed:

```
sudo apt-get install python2.7-dev python-numpy python-scipy python-h5py
python-dev python-tk python-mlt python-setuptools python-opengl python-qt4
python-vtk6 python-matplotlib python-openpyxl
```

# 3 Theory and code description

## 3.1 Motivation and theory behind the code

Relaxation zones can reduce undesired wave reflections at boundaries of the computational domain. However, relaxation zones contain user-defined parameters; these parameters are case-dependent and must be tuned for every simulation. Otherwise strong reflection may occur.

The present code can guide the tuning of these case-dependent parameters. The code is an implementation of the theory from [1]. Please see [1] to find out if and how the theory applies to the relaxation zone implementation that you use.

In [1], a theory was presented which predicts the reflection coefficient for a given relaxation zone setup. The reflection coefficient $C_\mathrm{R}$ 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.*

The theory predictions were demonstrated to be of satisfactory accuracy for regular long-crested waves of steepness up to $\approx 70\%$ of breaking steepness in both shallow and deep water, and also for 3D flows with strongly reflecting bodies in steep deep-water waves[1].

Future research is necessary to better understand how accurate the theory predicts the damping of short-crested waves, irregular waves and highly non-linear waves, such as rogue waves, waves close to breaking steepness and even breaking waves; literature results are promising that the theory covers these cases as well [1, 2, 3, 4, 5, 6].

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.

### 3.2 Introduction to relaxation zones

Relaxation zones blend, say a general transport equation $\mathcal{T}$ for transport quantity $\phi$, over to a reference solution via

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

where $b(\tilde{x})$ is a blending function such as e.g. Eqs. (4) to (6), $\mathcal{T}$ corresponds e.g. to the conservation equations for mass or momentum (without relaxation source terms), and $\mathcal{R}$ corresponds to $\int_V (\phi - \phi_{\mathrm{ref}})\ \mathrm{d}V$ with reference solution $\phi_{\mathrm{ref}}$ for transport quantity $\phi$. The relaxation parameter $\tau$ has unit [s] and regulates the magnitude of the source term in such a way that a large value of $\tau$ implicates a small source term and vice versa.

Hence the conservation equations for momentum and volume fraction take the form

$$\begin{aligned}(1 - b(\tilde{x}))\Bigg[&\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho u_i\ \mathrm{d}V + \int_S \rho u_i(\mathbf{v} - \mathbf{v}_g)\cdot\mathbf{n}\ \mathrm{d}S \\ &- \int_S (\tau_{ij}\mathbf{i}_j - p\mathbf{i}_i)\cdot\mathbf{n}\ \mathrm{d}S - \int_V \rho\mathbf{g}\cdot\mathbf{i}_i\ \mathrm{d}V\Bigg] \\ &+ \frac{b(\tilde{x})}{\tau}\Bigg[\int_V \rho\,(u_i - u_{i,\mathrm{ref}})\ \mathrm{d}V\Bigg] = 0 \quad ,\end{aligned} \tag{2}$$

$$\begin{aligned}(1 - b(\tilde{x}))\Bigg[&\frac{\mathrm{d}}{\mathrm{d}t}\int_V \alpha\ \mathrm{d}V + \int_S \alpha(\mathbf{v} - \mathbf{v}_{\mathrm{g}})\cdot\mathbf{n}\ \mathrm{d}S\Bigg] \\ &+ \frac{b(\tilde{x})}{\tau}\Bigg[\int_V (\alpha_i - \alpha_{i,\mathrm{ref}})\ \mathrm{d}V\Bigg] = 0 \quad ,\end{aligned} \tag{3}$$

with volume $V$ of control volume (CV) bounded by the closed surface S, fluid velocity $\mathbf{v}= (u_1, u_2, u_3)^{\mathrm{T}} = (u, v, w)^{\mathrm{T}}$, grid velocity $\mathbf{v}_g$, unit vector $\mathbf{n}$ normal to $S$ and pointing outwards, time $t$, pressure $p$, fluid density $\rho$, components $\tau_{ij}$ of the viscous stress tensor, unit vector $\mathbf{i}_j$ in direction $x_j$, and volume fraction $\alpha$ of the liquid phase, reference velocities $u_{i,\mathrm{ref}}$ and reference volume fraction $\alpha_{i,\mathrm{ref}}$, and blending functions such as e.g.

$$b(\tilde{x}) = \left(\frac{e^{((x_{\mathrm{d}} - \tilde{x})/x_{\mathrm{d}})^n} - 1}{e^1 - 1}\right) \quad , \tag{4}$$

$$b(\tilde{x}) = \left[\cos^2\left(\frac{\pi}{2} + \frac{\pi}{2}\left(\frac{x_{\mathrm{d}} - \tilde{x}}{x_{\mathrm{d}}}\right)\right)\right]^n \quad , \tag{5}$$

$$b(\tilde{x}) = \left(\frac{x_{\mathrm{d}} - \tilde{x}}{x_{\mathrm{d}}}\right)^n \quad , \tag{6}$$

where $\tilde{x}$ is the shortest distance to the closest domain boundary to which a relaxation zone of thickness $x_{\mathrm{d}}$ is attached, and $n$ regulates the shape of the blending function as illustrated in Fig. 1.

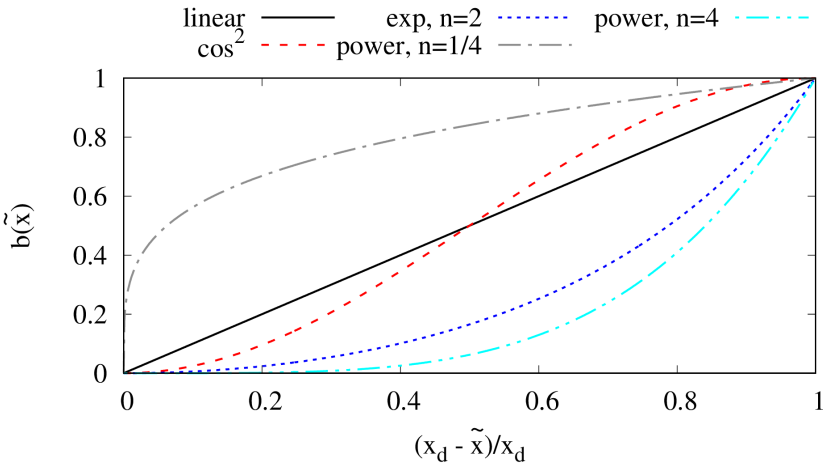



**Figure 1:** Different blending functions $b(\tilde{x})$ over location in relaxation zone, with entrance to the relaxation zones at $\frac{x_\mathrm{d}-\tilde{x}}{x_\mathrm{d}} = 0$ and boundary to which the relaxation zone is attached at $\frac{x_\mathrm{d}-\tilde{x}}{x_\mathrm{d}} = 1$; for blending via Eq. (6) with $n = 1$ ('linear'), $n = 1/4$ ('power, n=1/4') and $n = 4$ ('power, n=4'), blending via Eq. (5) with $n = 1$ ('$\cos^2$, n=1'), and blending via Eq. (4) with $n = 2$ ('exp, n=2')

Thus relaxation zones have three user-defined parameters, which are case-dependent and have to be adjusted for each simulation: the relaxation parameter $\tau$, the blending function $b(\tilde{x})$ and the zone thickness $x_\mathrm{d}$.

The difference between forcing zones and relaxation zones is that relaxation zones 'blend out' all terms except the source terms in the governing equations via the factor $(1 - b(\tilde{x}))$; forcing zones do not have this factor, thus with forcing zones the whole governing equations remain active in the whole domain, whereas within relaxation zones as in Eqs. (2) and (3), the terms from the governing equations that are active in the solution domain of interest are faded out and the reference solution is faded in.

Since the theory for predicting reflection coefficients for forcing zones [2] could be extended to relaxation zones in [1], one can conclude that forcing zones and relaxation zones are closely related. The results suggest that both approaches perform equally well, so the choice for using relaxation or forcing zones can be made based on which one is already implemented or easier to implement in one's flow solver.

In contrast to forcing zones, where higher order blending functions were found to perform better than linear or constant blending, for relaxation zones this is different: For example, constant blending with $b(\tilde{x}) = 1$ should be avoided, since then the entrance to the relaxation zone would behave like a velocity boundary condition, i.e. may be fully reflective. Consider that the theory for predicting relaxation zone behavior was derived by reformulating Eqs. (2) and (3) as a forcing zone, which was done be multiplying both equations with $(1-b(\tilde{x}))^{-1}$. Therefore in relaxation zones, the simultaneous blending-out of one solution (the parts in Eqs. (2) and (3) which are multiplied by $(1-b(\tilde{x}))$) while blending-in another solution (the parts in Eqs. (2) and (3) which are multiplied by $\frac{b(\tilde{x})}{\tau}$) can be interpreted as a forcing 'with a different blending', so the "effective" increase in source term magnitude is not directly proportional to $b(\tilde{x})$, but rather to $b(\tilde{x})/\left(1 - b(\tilde{x})\right)$. The optimum choice of blending $b(\tilde{x})$ was found to depend on the zone thickness $x_\mathrm{d}$. Since the dependence of reflection coefficient $C_\mathrm{R}$ on the case-dependent parameters is quite complex, the theory was found to be a useful tool for tuning the parameters. See [1] for a detailed discussion.

### 3.3 Benefits and limitations

The code gave satisfactory predictions for minimizing undesired wave reflections with regular waves of different steepness in shallow and deep water; due to the close relation to forcing zones, applicability to irregular waves can be expected. Further test-cases with short-crested waves are under investigation and will be published in due time.

Although the code's predictions are often quite accurate [2], please keep in mind that every theory has its limitations! For highly non-linear waves, such as breaking waves, or for oblique wave incidence in 3D, reflection may be larger than predicted. Further, undesired reflections can be due other mechanisms as well, e.g. the use of inappropriate grids. Therefore, tuning the relaxation zone parameters according to the present theory does NOT guarantee that the actual reflection coefficient in the simulation will equal the prediction. See [2] and [1] for a detailed discussion.

### 3.4 Recommendations for tuning relaxation zones

At the wave-maker, it is usually known which waves are generated. For a given wave period $T$, the wavelength $\lambda$ depends on the water depth and on the wave steepness, thus this has to be entered in the code. In practice, a relaxation zone thickness of $1\lambda \leq x_\mathrm{d} \leq 2\lambda$ is usually satisfactory. With correct tuning thinner zones may be used; in practice however, usually confidence in the minimization of undesired wave reflections is more important than the last few percent efficiency. Therefore it is recommended to use zone thicknesses $x_\mathrm{d}$ slightly larger than possibly necessary.

Increasing the zone thickness $x_\mathrm{d}$ widens the range of wavelengths which will be damped satisfactorily, and lowers the reflection coefficient for the optimum setting; thus if the wave absorption is not satisfactory, then the zone thickness $x_\mathrm{d}$ should be increased.

For irregular waves, a quick approach is to tune the zone to the peak period or the longest period[3]. A more accurate approach is to calculate the reflection coefficient for each wave component's period and in this manner tune the zone parameters accordingly.

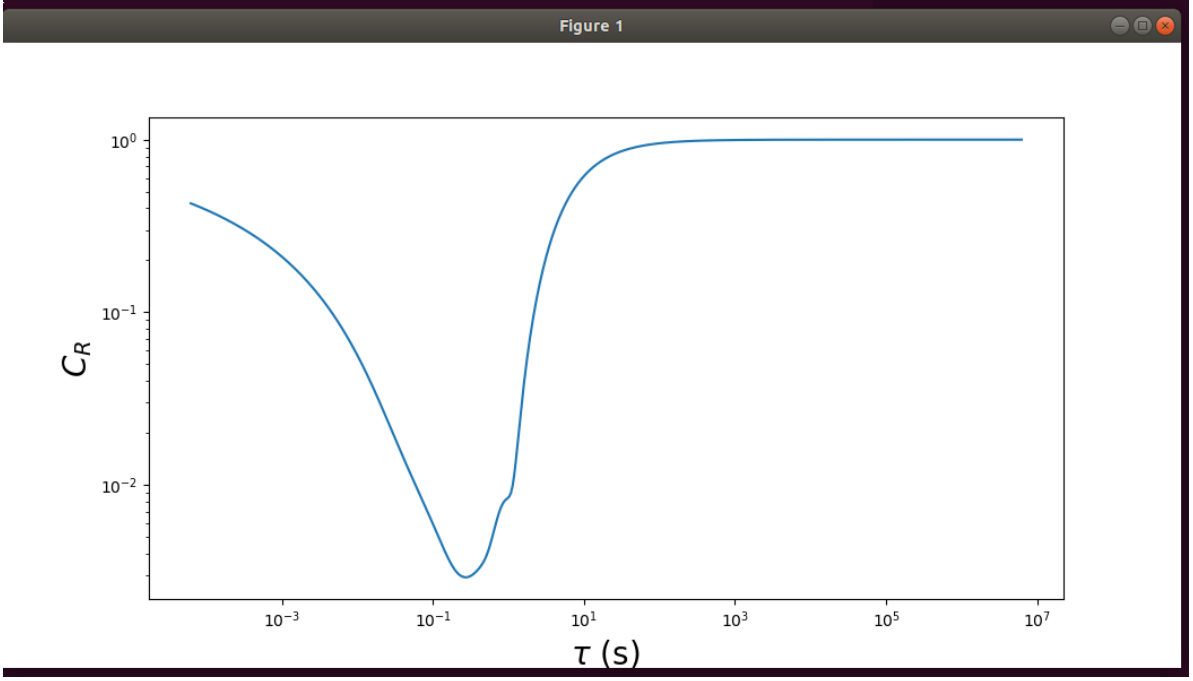
## 4 How to run the code

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

In Linux and macOS, open a terminal and type:
python CRestRelax.py

Example output:

```
python CRestRelax.py

Please enter wave period (s):
1.6

Please enter wavelength (m):
4

Please enter layer thickness per wavelength (enter 2 if thickness = 2 * wavelength):
2

Available blending functions are:
Power blending [b(x)=x**n]: 1
Cosine**2 blending [b(x)=(cos(x)**2)**n]: 2
Exponential blending [b(x)=(e**(x**n)-1)/(e**1-1)]: 3
Default setting Naval Hydro Pack is: exp

Please enter number of blending function:
3

Please enter power n:
3.5
```




# 5 Reporting bugs

So far, no bugs are known.

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

M.Sc. Robinson Peric
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/relaxation-zones-for-free-surface-waves`
Updates and further useful information will be posted there as well.

# 6 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 to set up their relaxation zones.

# 7 References


[1] R. Perić, V. Vukčević, M. Abdel-Maksoud, H. Jasak. Tuning the Case-Dependent Parameters of Relaxation Zones for Flow Simulations With Strongly Reflecting Bodies in Free-Surface Waves. Preprint, arXiv:????????? [physics.flu-dyn], 2018.

[2] 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.

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

[4] M. Perić. 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, 2015.

[5] R. Perić, M. Abdel-Maksoud. 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, 2015.

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