

# **Finding the correct implementation of Goda/Sommerfeld solutions on diffraction diagrams for directional random waves**

Final version

1749/U14094/C/HTAL

10 July 2014

SWAS

HYDRAULICS

COASTAL, HARBOUR AND RIVER CONSULTANTS

Schiehaven 13G

3024 EC Rotterdam

The Netherlands

☎ +31 - 10 - 467 13 61

☎ +31 - 10 - 467 45 59

✉ info@svasek.com

🌐 www.svasek.com

Document title Finding the correct implementation of  
Goda/Sommerfeld solutions on diffraction diagrams  
for directional random waves

Short document title Implementation of Goda solutions

Status Final

Date 10 July 2014

Project name Project

Project number 1749

Client R. Slomp (Rijkswaterstaat, Ministry of Infrastructure  
and the Environment), G. Van Vledder

Reference 1749/U14094/C/HTAL

Author Harmen Talstra

Checked by Pol van de Rest

## SAMENVATTING IN HET NEDERLANDS

**(English summary is given by Chapters 1 and 6 of this report)**

In de civieltechnische praktijk worden regelmatig diffractiediagrammen gebruikt; met behulp van deze diagrammen kan worden geschat hoe gevoelig havenbekkens en andere zones langs de kust zijn voor golfslag ten gevolge van golfdiffractie. Al sinds het invloedrijke werk van Penney en Price (1952) worden deze diffractiediagrammen gebaseerd op de befaamde analytische Sommerfeld-oplossing (1896) voor de diffractie van lichtgolven. Ondanks vele latere wetenschappelijke ontwikkelingen, met name op het gebied van de numerieke wiskunde, vormen standaard-diffractiediagrammen nog altijd een snel en praktisch instrument voor ontwerpdoeleinden.

Oorspronkelijk werden diffractiediagrammen alleen voor zuiver sinusvormige, regelmatige golven gemaakt. Door Goda c.s. (1978) is beschreven hoe spectrale informatie betreffende golfhoogte en golfrichting aan deze diffractiediagrammen kan worden toegevoegd. Deze zogeheten Goda-diagrammen beschrijven de diffractie van onregelmatige zee- en oceaangolven rond een aantal geschematiseerde havengeometrieën: een half-oneindige golfbreker en een havenmond tussen twee golfbrekers die in elkaars verlengde liggen (waarbij de havenmond uiteenlopende afmetingen kan hebben). Deze Goda-diagrammen liggen aan de basis van de zogeheten “spreadsheet-methode” (RIKZ, 2004), die in Nederland bij Rijkswaterstaat en waterschappen nog altijd in gebruik is om golfeffecten in havenbekkens af te schatten. Onderliggende aannames bij de Goda-diagrammen zijn: oneindig dunne golfbrekers met verticale wanden, een vlakke bodem, geen wrijving, geen wind, en geen transmissie van golfenergie door de golfbreker heen.

In hun huidige vorm hebben de Goda-diagrammen enkele nadelen:

- De huidige nauwkeurigheid van slechts één decimaal (overeenkomend met 10 cm golfhoogte) is soms onvoldoende en geeft aanleiding tot afrondingsfouten;
- Het is niet altijd duidelijk of afstanden in Goda-diagrammen dimensieloos gemaakt zijn met de breedte van de havenmond of met de golflengte. Deze verwarrende situatie kan aanleiding geven tot fouten;
- De invloed van de waterdiepte wordt niet behandeld door Goda c.s. (1978). Vermoedelijk hebben de auteurs abusievelijk de aanname gedaan dat het gebruik van diepwater-golfcondities altijd conservatieve golfhoogten oplevert;
- De oorspronkelijke Goda-diagrammen gaan uit van volledige golfreflectie langs de gehele binnen- en buitenzijde van de golfbreker(s). Deze aanname is niet nodig (partiële reflectie kan wiskundig ook goed behandeld worden) en kan leiden tot onnauwkeurige resultaten. Het kan wenselijk zijn om Goda-diagrammen met willekeurige reflectiecoëfficiënt te kunnen construeren.

Het doel van de onderhavige studie is om de achterliggende wiskundige methode achter de Goda-diagrammen opnieuw af te leiden en te implementeren, zodat de Goda-diagrammen “met één druk op de knop” kunnen worden gereconstrueerd en waar nodig aangepast aan de wensen van de gebruiker.

Let wel: het doel van deze studie is niet om het toepassingsgebied van de Goda-diagrammen te vergroten. Immers, aan de Goda-methode ligt een aantal aannames en vereenvoudigingen ten grondslag, die ook bij vergroting van de nauwkeurigheid van de methode van toepassing blijven. Het is daarom aan de gebruiker om te bepalen voor welke praktijkgevallen de Goda-methode kan worden toegepast. Voor gevallen waarbij bijv. niet-lineariteiten een belangrijke rol zullen spelen, is het gebruik van een state-of-the-art korte-golfmodel vereist. Wel kunnen de nieuwe, meer verfijnde Goda-oplossingen uiteraard goed gebruikt worden om een dergelijk state-of-the-art model te valideren.

Hoewel Goda c.s. (1978) de wiskunde achter hun diagrammen deels beschrijven, is deze beschrijving onvolledig. Er lijkt sprake te zijn van een aantal verloren gegane “missing links” in de wiskundige

kennis om de Goda-diagrammen te kunnen afleiden, ook in de overige beschikbare literatuur. In de huidige studie worden deze “missing links” op aannemelijke wijze achterhaald en wordt de gehele methode van A tot Z afdoende gedocumenteerd.

In deze studie wordt de methode van Goda c.s. (1978) stap voor stap afgeleid en waar nodig achterhaald. Het eindproduct bestaat uit een computermodel (geprogrammeerd in de vorm van een reeks Matlab-scripts) waarmee de diffractiediagrammen van Goda kunnen worden gereproduceerd, d.w.z. de originele diagrammen of, indien gewenst, door de gebruiker gedefinieerde variaties op dit hoofdthema. Dit stelt de gebruiker in staat om diffractiediagrammen te maken:

- Met veel grotere nauwkeurigheid (meer dan 1 decimaal);
- Met iedere gewenste ruimtelijke resolutie;
- Voor iedere gewenste golfbrekeroriëntatie en reflectiecoëfficiënt;
- Voor iedere gewenste golfrichting (daar waar de oorspronkelijke Goda-diagrammen slechts uitgingen van loodrechte golfinval);
- Reflectiecoëfficiënten, waterdiepte en golfspectrum kunnen vrijelijk worden aangepast.

Mogelijke aanpassingen die de gebruiker aan de Goda-diagrammen kan maken zijn:

- Een goed gekalibreerde reflectiecoëfficiënt (tussen de 0.0 en 1.0);
- Het dimensieloos maken van de diffractiediagrammen met behulp van de golflengte of andere horizontale lengteschalen zoals de havenmondbreedte, al naar gelang de voorkeur van de gebruiker;
- Variatie van de waterdiepte (globaal, dus niet ruimtelijk);
- Aanpassing van het golfenergiespectrum dat in de Matlab-scripts is geïmplementeerd. Ieder fysisch relevant type golfenergiespectrum is hiervoor in principe bruikbaar.

De diffractietabellen van de “spreadsheet-methode” (RIKZ, 2004) zijn met behulp van de ontwikkelde Matlab-scripts gereproduceerd. Het enige verschil met de originele tabellen is dat de nieuwe tabellen zijn bepaald met een grotere nauwkeurigheid (2 in plaats van 1 decimaal achter de komma) en vermoedelijk ook een hogere spectrale resolutie. Daarnaast lijken de originele waarden in de tabellen van de “spreadsheet-methode” te zijn bepaald door aflezing van waarden in de originele figuren uit Goda et al. (1978), terwijl de nieuwe tabellen direct zijn berekend met behulp van de output van de Matlab-scripts. Eveneens zijn de tabellen nu reproduceerbaar en het is duidelijk geworden welke aannamen zijn gedaan. De nieuwe tabellen zijn verwerkt in de Excel-sheet “Diffractiediagrammen\_GODA\_09072014.xlsx”.

## TABLE OF CONTENTS

	Pag.
SAMENVATTING IN HET NEDERLANDS	I
1 INTRODUCTION	1
2 SETUP OF THE STUDY	3
3 FIRST STEP: SOMMERFELD SOLUTION FOR SEMI-INFINITE BREAKWATER	4
3.1 Basic equations	4
3.2 Sommerfeld solution and Fresnel integrals	6
3.3 Correction for wave directions outside the domain $0 \leq \theta_0 \leq \pi$	6
3.4 Comparison of Sommerfeld solution formulation with Goda formulation	7
3.5 Explanation of correction for wave direction angles	9
3.6 Extension to arbitrary breakwater orientation	9
3.7 Implementation	11
4 SECOND STEP: APPROXIMATE SOLUTION FOR BREAKWATER GAP	13
4.1 Introduction: a re-construction strategy	13
4.2 Satisfaction of Helmholtz Equation and boundary conditions	13
4.3 “Right half” and “left half” of the breakwater gap solution	19
4.4 Solution assembly: approximate solution of the breakwater gap problem	20
4.5 Assembled solution: approximate satisfaction of boundary conditions	23
4.6 Implementation	28
5 THIRD STEP: ADDING SPECTRAL AND DIRECTIONAL SPREADING CHARACTERISTICS	30
5.1 Introduction	30
5.2 Frequency spectrum; significant wave height and period	30
5.3 Directional spectrum; correct integration over wave directions	32
5.4 Spectral calculation of wave diffraction	34
5.5 Implementation	36
6 SUMMARY AND CONCLUSIONS	38
REFERENCES	40
APPENDICES	I
Appendix A: Matlab script SemiInfiniteBreakwater_Test.m	i
Appendix B: Matlab function Sommerfeld_PenneyPrice.m	iv
Appendix C: Matlab function Sommerfeld_Goda.m	v
Appendix D: Matlab script CornuSpiral.m	vi
Appendix E: Matlab script BreakwaterGap_Test.m	vii
Appendix F: Matlab script BoundaryConditions_Gap.m	xi
Appendix G: Matlab script SpectralIntegral.m	xiv
Appendix H: Matlab function WaveNumber.m	xvii
Appendix I: Matlab script GodaDiagram_SemiInfiniteBreakwater.m	xviii
Appendix J: Matlab script GodaDiagram_BreakwaterGap.m	xxi

## 1 INTRODUCTION

Diffraction diagrams are a tool frequently used in civil engineering practice, which provide an estimate of the sensitivity of harbour basins (and other coastal areas with breakwaters) to wave influence due to diffraction. Ever since the principal proof by Penney and Price (1952) that the diffraction of water waves by breakwaters can be described (for some standard cases) with Sommerfeld's (1896) analytical solution for the diffraction of light, wave heights behind breakwaters have been often estimated with the aid of several standard diffraction diagrams. Although much more dedicated (and more advanced) alternative approaches for wave diffraction have been developed since the 1950s (e.g. numerical computer models based on the Mild-Slope Equation, Boussinesq-type approximations or even the full 3D Navier-Stokes Equations), standard diffraction diagrams are still in use for design purposes in order to quickly provide a first estimate.

Initially, such diffraction diagrams did only account for monochromatic waves: completely regular waves of a single wave length from a single direction. It has been described by Goda et al. (1978) how spectral information (frequency spreading and directional spreading) could be added to these monochromatic wave diagrams, leading to the so-called Goda diffraction diagrams, which describe the diffraction of directional random waves around a semi-infinite breakwater, resp. a gap between two breakwaters. These Goda diagrams are currently still in use, e.g. within design guidelines by the Dutch Ministry of Public Works (RIKZ, 2004) for the estimation of wave impact in harbour areas. These guidelines, otherwise known as the "spreadsheet method", uses the original diffraction diagrams by Goda in table form, with an accuracy of only 1 decimal place (sometimes 2). Basic assumptions underlying to these Goda diagrams are:

- Breakwaters are infinitely thin (vertical walls);
- Flat bottom;
- No friction and dissipation effects;
- No wind or other meteorological effects;
- No transmission of wave energy over or throughout the breakwater;
- Both breakwaters for the breakwater gap case are aligned with each other.

In their present form, the Goda diagrams as used for design purposes have several drawbacks:

- The current accuracy of only 1 decimal place, on quite a coarse two-dimensional horizontal grid of points, sometimes appears to be insufficient and may easily give rise to significant rounding errors;
- In the original diffraction diagrams by Goda, the horizontal coordinates  $x$  and  $y$  (in the region behind the breakwater) have been normalized by the breakwater gap width  $B$ ; in the spreadsheet method, however, these coordinates have been normalized by the wave length  $L$  instead. This may easily lead to confusion and even to a totally incorrect application of the method;
- Strikingly, Goda et al. (1978) do not mention the influence of the water depth at all. Presumably, the original authors did assume that applying deep-water wave conditions would lead to conservative results, which may not always be true. Within near-shore harbours, the influence of limited water depth on incident waves can be expected to be important – and even more if also frequency spreading is present;
- It has been found by Boshek (2009) that the original diffraction diagrams by Goda must have been constructed under the assumption of a fully reflecting breakwater, both at the sea-side and the lee-side (reflection coefficient  $R = 1$ ). Although this is a conservative assumption, it is not sufficiently accurate under all circumstances. Strictly spoken, the mathematical method adopted by Goda et al. (1978) allows for choosing any reflection coefficient  $R$  between 0.0 and 1.0. It would be desirable if "customized" diffraction diagrams with a proper choice for the reflection coefficient  $R$  could be quickly generated whenever necessary.

In view of these drawbacks, we aim to “redo” the work done by Goda et al. (1978), in such a way that a larger accuracy is achieved and that “customizations on demand” can be produced on the spot. Such customizations could include, for instance:

- Choosing a well-tuned reflection coefficient between 0.0 and 1.0;
- Normalizing the diffraction diagrams in horizontal direction by either L or B, or any other horizontal length scale;
- Variation of the water depth (not spatially);
- Variation of the variance density spectrum and directional spectrum of the random waves;
- Replacing the analytical Sommerfeld solution by a numerical solution, like e.g. a solution of the Mild-Slope Equations (for validation purposes).

Apparently, the original knowledge of how to actually *implement* the analytical solutions provided by Goda et al. (1978) to find Goda’s diffraction diagrams (as utilized within the spreadsheet method) has been lost long ago. The original 1978 paper covers most of the analytical aspects of the Goda method, yet not all of them; some implementation aspects are not explicitly clarified therein (nor in other available literature) and must be examined thoroughly.

The objective of this study is to implement the method by Goda et al. (1978) step by step, and possibly to re-invent some of these steps. These steps will provide a tool that is able to reproduce the Goda diagrams, i.e. both the original diagrams and any customized variation of it, at sufficient accuracy. We provide a proper documentation of all steps needed to construct these diagrams, as such a description of implementation, though certainly relevant for engineers, did not seem to exist before now.

The results of this study enable us to produce Goda diagrams:

- At much larger accuracy (much more than only 1 decimal place);
- At any desirable spatial resolution;
- For any breakwater orientation and reflection coefficient (though only approximately for the breakwater gap case);
- For any dominant wave direction, whereas the original Goda diagrams only accounted for normal wave incidence;
- Reflection coefficients, water depth and wave spectra can be freely adapted.

It should be noted that enlarging the applicability range of the Goda method is not an objective of this study. The basis of the Goda method includes a number of assumptions and simplifications, which do not vanish if the accuracy of the method is enhanced. The model user should be aware of the assumptions behind the Goda method and should decide to which practical cases the method may be applied. For example, cases in which non-linearities are expected to play an important role will require the use of a state-of-the-art wave model. On the other hand, the current new (and more accurate) Goda solutions are obviously useful for the validation of such a state-of-the-art models.

Chapter 2 describes the setup of this study. Three steps are distinguished which must be taken to derive and re-implement the Goda diagrams. Chapters 3, 4 and 5 each address one of these steps; at the end of each chapter, the achieved results are discussed and demonstrated with the help of several Matlab scripts. Finally Chapter 6 offers a summary and conclusions, followed by references.

## 2 SETUP OF THE STUDY

The basis of the diffraction diagrams constructed in the past is given by Sommerfeld's (1896) analytical solution for the (linear) wave field around a semi-infinitely long and infinitesimally thin breakwater, at constant water depth. The first step we must take is to implement this analytical solution, in such a way that it is valid and robust for all wave directions and for any orientation of the breakwater. Penney and Price (1952) and Boshek (2009) give formulations for the Sommerfeld solution that are slightly different from Goda et al. (1978); one of our purposes is to unify them.

The Sommerfeld solution is *exact* for the case of a single semi-infinite breakwater which extends to infinity from the origin along the positive x-axis. It is only *approximately* correct for the case of two breakwaters aligned with each other, with a gap in between that is at least larger than one wavelength. Our second step is to use the semi-infinite breakwater solution to construct an approximate solution for the breakwater gap case. This step is *not* described explicitly by Goda et al. (1978), although it has been a vital step for them to construct the Goda diffraction diagrams (!). Neither has it been described explicitly within any other literature that has been found so far, although some local analytical approximations have been given by Penney and Price (1952).

Having constructed the analytical solutions above for the case of monochromatic waves, the third step is to add spectral and directional information to these solutions. This step has been reasonably well described by Goda et al., although the way in which spectra must be integrated along the directional dimension can be subject to some discussion. In the present study, some adaptations to the implementation are made where necessary.

All steps described here have been implemented in the form of Matlab .M-file scripts with sufficient comment lines. These Matlab scripts are given at the end of this report (in the form of Appendices).



### 3 FIRST STEP: SOMMERFELD SOLUTION FOR SEMI-INFINITE BREAKWATER

#### 3.1 Basic equations

Mathematically, the two-dimensional diffraction of light waves (resp. water waves) around a semi-infinite screen (resp. breakwater, at *constant* water depth) is described by the two-dimensional Helmholtz problem:

$$\nabla^2 F + k^2 F = 0, \quad (1)$$

where  $k$  is the wave number and  $F$  is the (complex) horizontal part of the 3D wave velocity potential function (from linear wave theory). Alternatively, the function  $F$  may be replaced by the complex surface elevation amplitude  $\zeta$ ; although there is a phase difference of  $\frac{1}{2}\pi$  between  $\zeta$  and  $F$ , Equation (1) remains valid in that form. It is also possible to interpret the modulus of  $F$  as a diffraction coefficient related to an incident wave with unit height; in that case, the absolute value of the computed field  $F$  must be multiplied by the actual incident wave amplitude (or wave height) in order to find the final solution.

For a semi-infinite breakwater along the positive x-axis, the following boundary condition holds:

$$\frac{\partial F}{\partial y} = 0 \quad \text{at } y = 0, x \geq 0 \quad (2)$$

Sommerfeld (1896) solved the system (1)-(2) analytically. The full description of his analytical solution applied to breakwaters was first given by Penney and Price (1952). This solution is usually given in polar coordinates:

$$r = \sqrt{x^2 + y^2}, \quad \theta = \arctan(y, x) \quad (3)$$

N.B.: It turns out to be important that the values of  $\theta$  are chosen within the range  $0 \leq \theta \leq 2\pi$ , and *not* within  $-\pi \leq \theta \leq \pi$ . Obviously, the location of the breakwater is given by  $\theta = 0$ . Furthermore we define, in accordance with Penney and Price, the wave direction  $\theta_0$  as the direction that the undisturbed wave train is heading *toward*. In other words,  $\theta_0$  is the direction of the "shadow line" of the wave train behind the breakwater, as seen from the origin (see Figure 1).

The Sommerfeld solution  $F$  consists of two parts, referring to the incident wave ( $F_i$ ) and the reflected wave ( $F_r$ ) respectively:

$$F = F_i + R \cdot F_r \quad (4)$$

The reflection coefficient  $R$  may range between 0.0 and 1.0. Both parts of the solutions contain a region with an undisturbed wave and a region with a scattered (diffracted) wave. The incident wave and reflected wave with their respective scatter fields, as well as the total Sommerfeld solution, are depicted in Figure 2 for one particular wave direction. In Figure 1, three characteristic regions within the solution field have been indicated: the regions Q, R and S. The region R contains mainly incident and reflected waves; the region S contains scattered waves only; and the intermediate region Q is dominated by both scattered and incident waves. The boundaries between these regions are given by the shadow lines of the incident wave and the reflected wave, i.e. in polar coordinates the lines  $\theta = \theta_0$  and  $\theta = 2\pi - \theta_0$ .

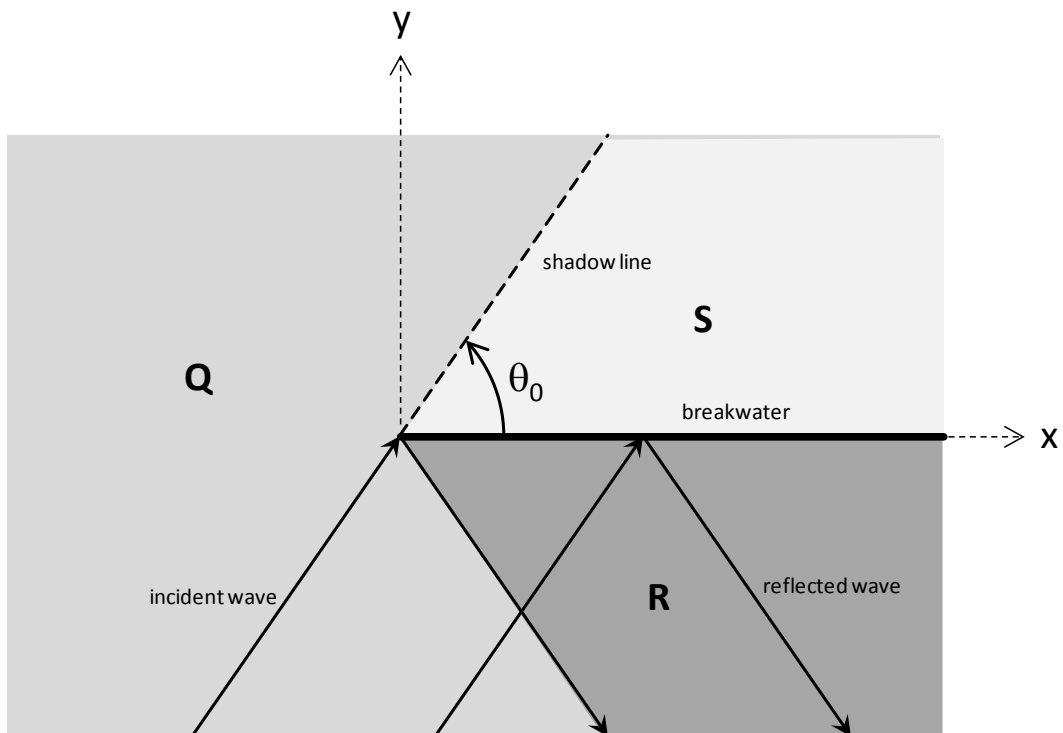


Figure 1: Setup and notation for the various regions of the Sommerfeld solution.

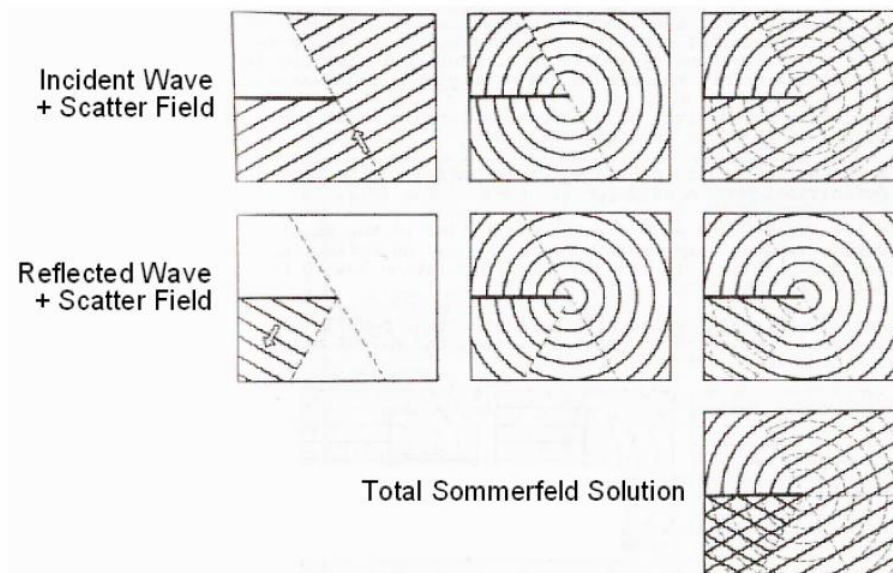


Figure 2: Superposition of partial wave fields within Sommerfeld solution (Daemrich, 1978). Upper row: “undisturbed” incident wave (left), scattered incident wave (middle), and the sum of both (right). Middle row: “undisturbed” reflected wave (left), scattered reflected wave (middle), and the sum of both (right). Lower row: total sum all partial wave fields according to Sommerfeld.

### 3.2 Sommerfeld solution and Fresnel integrals

The Sommerfeld solutions of the partial waves  $F_i$  and  $F_r$ , given by Penney and Price (1952), are:

$$F_i = \left\{ \frac{1}{2} + f(\sigma_i) \right\} \exp \{ -ikr \cos(\theta - \theta_0) \} \quad \text{and}$$

$$F_r = \left\{ \frac{1}{2} + f(\sigma_r) \right\} \exp \{ -ikr \cos(\theta + \theta_0) \}, \quad (5)$$

Where the function  $f(\sigma_i)$  and  $f(\sigma_r)$  are complex Fresnel integrals:

$$f(\sigma) = \frac{1+i}{2} \int_0^\sigma \exp \left\{ -i \frac{1}{2} \pi u^2 \right\} du \quad (6)$$

Basically, the Fresnel integral is a continuous summation of point sources along a line (according to Huygens' principle of wave propagation). Plotted in the complex plane for all values of  $\sigma$  (from  $-\infty$  to  $+\infty$ ), the Fresnel integral forms the well-known double-sided Cornu spiral. Because

$$\frac{1+i}{2} \int_0^\infty \exp \left\{ -i \frac{1}{2} \pi u^2 \right\} du = \frac{1+i}{2} \int_{-\infty}^0 \exp \left\{ -i \frac{1}{2} \pi u^2 \right\} du = \frac{1}{2} \quad \text{and} \quad \frac{1+i}{2} \int_{-\infty}^\infty \exp \left\{ -i \frac{1}{2} \pi u^2 \right\} du = 1,$$

we can see from the first factor in Equation (5), including the extra term  $\frac{1}{2}$ , that the Fresnel integrals are basically integrated between  $-\infty$  and  $\sigma_i$  or  $\sigma_r$ . This is because there is no breakwater present on the left side of the domain which extends toward  $-\infty$ , whereas there is indeed a breakwater at the right side of the domain.

The integration boundaries  $\sigma_i$  and  $\sigma_r$  are given by:

$$\sigma_i = 2\sqrt{\frac{kr}{\pi}} \sin\left(\frac{\theta - \theta_0}{2}\right) \quad \text{and} \quad \sigma_r = -2\sqrt{\frac{kr}{\pi}} \sin\left(\frac{\theta + \theta_0}{2}\right) \quad (7)$$

As pointed out by Penney and Price (1952), the signs of  $\sigma_i$  and  $\sigma_r$  actually define the solution regions Q, R and S:

- Both  $\sigma_i$  and  $\sigma_r$  are positive in the reflection region R;
- Both  $\sigma_i$  and  $\sigma_r$  are negative in the scatter region S;
- $\sigma_i$  is positive and  $\sigma_r$  is negative in the intermediate region Q.

In general, a negative value of  $\sigma_i$  or  $\sigma_r$  indicates wave scatter and a positive value indicates more or less undisturbed wave propagation.

### 3.3 Correction for wave directions outside the domain $0 \leq \theta_0 \leq \pi$

The expressions (5)-(7) are given by both Penney and Price (1952) and Boshek (2009). However, it turns out that these expressions are not valid for *all* values of the wave direction  $\theta_0$ . This is caused by the fact that the expressions in Equation (7) contain a sine function of *half an angle*. This implies that, when the value  $2\pi$  is added to (or subtracted from) these angles, the expressions between

brackets in (7) will only increase or decrease by a value  $\pi$ , so that  $\sigma_i$  and  $\sigma_r$  will change sign (!). The expressions within equation (7) are therefore only valid for angles  $0 \leq \theta_0 \leq \pi$ . This is not robust. The problem can be easily solved by a correction factor for the sign of  $\sigma_i$  and  $\sigma_r$ . Therefore, we propose here:

$$\sigma_i = 2\sqrt{\frac{kr}{\pi}} \sin\left(\frac{\theta - \theta_0}{2}\right) \text{sign}\left\{\cos\left(\frac{\theta_0}{2}\right)\right\}, \quad \sigma_r = -2\sqrt{\frac{kr}{\pi}} \sin\left(\frac{\theta + \theta_0}{2}\right) \text{sign}\left\{\cos\left(\frac{\theta_0}{2}\right)\right\}, \quad (8)$$

Where the sign-function (according the Matlab convention) is given by:

$$\text{sign}(x) = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x = 0 \\ -1 & \text{for } x < 0 \end{cases} \quad (9)$$

Adopting the expressions in Equation (8), the Sommerfeld solution (5)-(6) is valid for any wave direction angle  $\theta_0$ : not only  $0 \leq \theta_0 \leq 2\pi$ , but also larger values and smaller (negative) values.

### 3.4 Comparison of Sommerfeld solution formulation with Goda formulation

Goda et al. (1978) give formulations for the Sommerfeld solution that are slightly different from Penney and Price (1952) and Boshek (2009). In this section we will show in which way they are equivalent, and correct the Goda solution as well in order to make it robust for any value of the wave direction angle.

The full solution for  $F_i$  and  $F_r$ , as denoted by Goda et al. (1978) reads:

$$\begin{aligned} F_i &= \frac{1}{\sqrt{2}} \exp\left[i\left\{kr \cos(\alpha - \theta) + \frac{\pi}{4}\right\}\right] \cdot \left[\left\{C(\gamma_1) + \frac{1}{2}\right\} - i\left\{S(\gamma_1) + \frac{1}{2}\right\}\right] \\ &= \exp\left[i\left\{kr \cos(\alpha - \theta)\right\}\right] + \frac{1}{\sqrt{2}} \exp\left[i\left\{kr \cos(\alpha - \theta) + \frac{\pi}{4}\right\}\right] \cdot \left[\left\{C(\gamma_1) - \frac{1}{2}\right\} - i\left\{S(\gamma_1) - \frac{1}{2}\right\}\right] \\ F_r &= \frac{1}{\sqrt{2}} \exp\left[i\left\{kr \cos(\alpha + \theta) + \frac{\pi}{4}\right\}\right] \cdot \left[\left\{C(\gamma_2) + \frac{1}{2}\right\} - i\left\{S(\gamma_2) + \frac{1}{2}\right\}\right] \\ &= \exp\left[i\left\{kr \cos(\alpha + \theta)\right\}\right] + \frac{1}{\sqrt{2}} \exp\left[i\left\{kr \cos(\alpha + \theta) + \frac{\pi}{4}\right\}\right] \cdot \left[\left\{C(\gamma_2) - \frac{1}{2}\right\} - i\left\{S(\gamma_2) - \frac{1}{2}\right\}\right] \end{aligned}$$

where:

$$\gamma_1 = \sqrt{\frac{4kr}{\pi}} \cos\left(\frac{\alpha - \theta}{2}\right) \quad \text{and} \quad \gamma_2 = \sqrt{\frac{4kr}{\pi}} \cos\left(\frac{\alpha + \theta}{2}\right),$$

and:

$$C(\gamma) = \int_0^\gamma \cos\left(\frac{1}{2}\pi u^2\right) du \quad \text{and} \quad S(\gamma) = \int_0^\gamma \sin\left(\frac{1}{2}\pi u^2\right) du \quad (10)$$

In fact, Goda et al. give two expressions for each partial wave. For the comparison with the Sommerfeld solution as given by Penney and Price (1952) and Boshek (2009), the second expression is not needed and can be discarded here. The following expressions remain:

$$F_i = \frac{1}{\sqrt{2}} \exp \left[ i \left\{ kr \cos(\alpha - \theta) + \frac{\pi}{4} \right\} \right] \cdot \left[ \left\{ C(\gamma_1) + \frac{1}{2} \right\} - i \left\{ S(\gamma_1) + \frac{1}{2} \right\} \right] \quad \text{and}$$

$$F_r = \frac{1}{\sqrt{2}} \exp \left[ i \left\{ kr \cos(\alpha + \theta) + \frac{\pi}{4} \right\} \right] \cdot \left[ \left\{ C(\gamma_2) + \frac{1}{2} \right\} - i \left\{ S(\gamma_2) + \frac{1}{2} \right\} \right] \quad (11)$$

First, we note that the definition of the wave direction angle is different from that of Penney and Price here. The wave direction  $\theta$  defined by Goda differs  $180^\circ$  from that of Penney and Price: it is actually the direction that the wave is *coming from*. The polar coordinate  $\alpha$  defined by Goda is equal to the polar coordinate  $\theta$  defined by Penney and Price. First, we substitute the Penney and Price angle definitions and notations into the Goda formulations. We shift the wave direction angle with a value  $+\pi$ . This yields:

$$F_i = \frac{1}{\sqrt{2}} \exp \left[ i \left\{ kr \cos(\theta - \theta_0 - \pi) + \frac{\pi}{4} \right\} \right] \cdot \left[ \left\{ C(\gamma_1) + \frac{1}{2} \right\} - i \left\{ S(\gamma_1) + \frac{1}{2} \right\} \right] \quad \text{and}$$

$$F_r = \frac{1}{\sqrt{2}} \exp \left[ i \left\{ kr \cos(\theta + \theta_0 + \pi) + \frac{\pi}{4} \right\} \right] \cdot \left[ \left\{ C(\gamma_2) + \frac{1}{2} \right\} - i \left\{ S(\gamma_2) + \frac{1}{2} \right\} \right],$$

where:

$$\gamma_1 = \sqrt{\frac{4kr}{\pi}} \cos \left( \frac{\theta - \theta_0 - \pi}{2} \right) \quad \text{and} \quad \gamma_2 = \sqrt{\frac{4kr}{\pi}} \cos \left( \frac{\theta + \theta_0 + \pi}{2} \right) \quad (12)$$

This can be reworked into:

$$F_i = \frac{1}{\sqrt{2}} \exp \left[ i \left\{ -kr \cos(\theta - \theta_0) + \frac{\pi}{4} \right\} \right] \cdot \left[ \left\{ C(\sigma_i) + \frac{1}{2} \right\} - i \left\{ S(\sigma_i) + \frac{1}{2} \right\} \right] \quad \text{and}$$

$$F_r = \frac{1}{\sqrt{2}} \exp \left[ i \left\{ -kr \cos(\theta + \theta_0) + \frac{\pi}{4} \right\} \right] \cdot \left[ \left\{ C(\sigma_r) + \frac{1}{2} \right\} - i \left\{ S(\sigma_r) + \frac{1}{2} \right\} \right],$$

where:

$$\gamma_1 = 2\sqrt{\frac{kr}{\pi}} \sin \left( \frac{\theta - \theta_0}{2} \right) = \sigma_i \quad \text{and} \quad \gamma_2 = -2\sqrt{\frac{kr}{\pi}} \sin \left( \frac{\theta + \theta_0}{2} \right) = \sigma_r \quad (13)$$

Hence, we may replace  $\gamma_1$  and  $\gamma_2$  by the corresponding integral boundaries  $\sigma_i$  and  $\sigma_r$ . However, if we would have shifted the wave direction angle with a value  $-\pi$ , we would have got  $\gamma_1 = -\sigma_i$  and  $\gamma_2 = -\sigma_r$  instead. This indicated that neither the Goda formulation nor the original Penney & Price formulation is valid for all wave directions, so that also the Goda formulation has to be corrected in some way.

Next, we note that the functions  $C(\sigma)$  and  $S(\sigma)$  are a form of the real and imaginary part of the complex Fresnel integral. Furthermore, comparing Equation (11) with (6), it can be seen that:

$$\frac{1}{\sqrt{2}} \exp\left[i \frac{\pi}{4}\right] = \frac{1+i}{2} \quad (14)$$

Because of this, we can rewrite Equation (13) in a much more compact form, using the definition of  $f(\sigma)$  in Equation (6):

$$F_i = \exp[-ikr \cos(\theta - \theta_0)] \cdot \left[ f(\sigma_i) + \frac{1}{2} \right] \text{ and}$$

$$F_r = \exp[-ikr \cos(\theta + \theta_0)] \cdot \left[ f(\sigma_r) + \frac{1}{2} \right] \quad (15)$$

These expressions are equal to Equation (5), i.e. the Goda and Penney & Price formulations of the Sommerfeld solution are equivalent.

When we return to the original Goda formulation (10), we state here that this formulation can be made robust (for any value of the wave direction angle) in the following way:

$$\gamma_1 = \sqrt{\frac{4kr}{\pi}} \cos\left(\frac{\alpha - \theta}{2}\right) \text{sign}\left\{\sin\left(\frac{\theta}{2}\right)\right\}, \quad \gamma_2 = \sqrt{\frac{4kr}{\pi}} \cos\left(\frac{\alpha - \theta}{2}\right) \text{sign}\left\{\sin\left(\frac{\theta}{2}\right)\right\}, \quad (16)$$

Thus both the Penney and Price (1952) and the Goda et al. (1978) formulations have been corrected now in order to make them valid for the entire range of wave direction angles.

### 3.5 Explanation of correction for wave direction angles

Having come so far, we may explain why this correction is needed whatsoever. The reason why the original values of  $\sigma_i$ ,  $\sigma_r$ ,  $\gamma_1$  and  $\gamma_2$  are not naturally valid for any wave direction, is that these four variables are in fact square roots of four other quantities, which represent a phase difference between point sources according to Huygens' principle. For the Penney and Price case, this phase difference expression contains a factor  $\{1 - \cos(\theta - \theta_0)\}$ , which is equal to  $2 \sin^2\left(\frac{1}{2}\theta - \frac{1}{2}\theta_0\right)$ . For the Goda case, the phase difference contains  $\{1 - \cos(\theta - \theta_0 - \pi)\} = \{1 + \cos(\theta - \theta_0)\}$ , which is actually equal to  $2 \cos^2\left(\frac{1}{2}\theta - \frac{1}{2}\theta_0\right)$ . Taking the square roots of these quadratic expressions, we find the formulations for  $\sigma_i$ ,  $\sigma_r$ ,  $\gamma_1$  and  $\gamma_2$  in Equations (7) and (10) respectively (apart from some possible minus signs which are needed to obtain the correct wave scatter behaviour). However, the sine of cosine of *half an angle* (due to the square root) is not periodic in  $2\pi$ , but in  $4\pi$  (!). This artifact needs to be corrected for by the correction factors proposed in Equations (8) and (16).

### 3.6 Extension to arbitrary breakwater orientation

From this point, having shown the equivalence of the Penney and Price (1952) and Goda (1978) formulations, we proceed with the (corrected) Penney and Price formulation which is given by Equations (5), (6) and (8). This formulation is valid for a breakwater along the positive x-axis, extending from the origin towards infinity. It can however easily be extended to a more general case, in which the breakwater head has an arbitrary location and may extend in an arbitrary direction. This generalization will be helpful later on, in order to easily derive the case of a semi-infinite breakwater extending *to the left* instead of to the right, as well as the case of a gap between two breakwaters.

We define a horizontal plane with coordinates  $(x, y)$  and origin  $O$ . The breakwater head is located at a point  $(x_0, y_0)$  and its orientation toward infinity is given by the angle  $\delta$  (see Figure 3). The global wave direction is given by the angle  $\beta$ . Now we define a local coordinate system  $(\bar{x}, \bar{y})$ , according to:

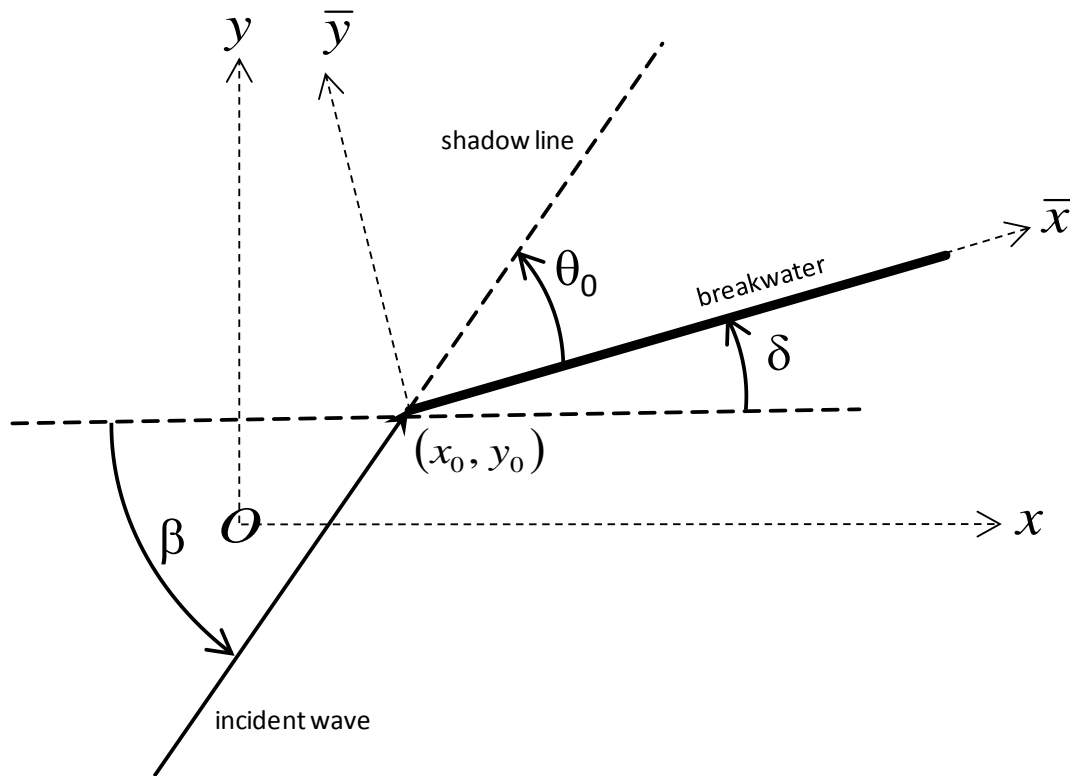


Figure 3: Global and local coordinate system for breakwater with arbitrary orientation.

$$\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = \begin{bmatrix} \cos \delta & \sin \delta \\ -\sin \delta & \cos \delta \end{bmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix} \quad (17)$$

The corresponding polar coordinate system is given by:

$$r = \sqrt{\bar{x}^2 + \bar{y}^2}, \quad \theta = \arctan(\bar{y}, \bar{x}), \quad (18)$$

with  $0 \leq \theta \leq 2\pi$  (!). The wave direction in the local system is given by:  $\theta_0 = \beta - \delta$ . Finally, the solution that we are looking for is simply given by Equations (5), (6) and (8).

If we define a default computation with a semi-infinite breakwater *to the right* with orientation  $\delta = 0$  and wave direction  $\theta_0 = \beta$ , we can find the corresponding case with a semi-infinite breakwater *to the left* by setting  $\delta = \pi$  and wave direction  $\theta_0 = \beta - \pi$ .

### 3.7 Implementation

The formulations outlined in the previous sections have been implemented within 4 Matlab scripts, which can be found in Appendices A-D.

The actual Sommerfeld model according to the description by Penney and Price (1952) has been implemented in the Matlab function [Sommerfeld PenneyPrice.m](#), which is listed in Appendix B. Appendix C contains a fully corresponding matlab function, named [Sommerfeld Goda.m](#), in which the Sommerfeld solution is implemented according to Goda et al. (1979). Both functions are tested by the Matlab script [SemilfiniteBreakwater Test.m](#), which is found in Appendix A. All these scripts make use of the simple Matlab data file [Cornu.mat](#), which contains vectors with values of  $\sigma$  and the Fresnel integral  $f(\sigma)$  at high resolution. This data file can be constructed by running the Matlab script [CornuSpiral.m](#), which is found in Appendix D. All Matlab files include sufficient comments to make them understandable for the reader (in combination with the information from the previous sections).

In principle, the difference between the output of the Matlab functions in Appendix B and C should be zero to machine precision; this can be checked by running the script within Appendix A. The only exception where the Sommerfeld methods according to Penney and Price versus Goda may yield a different result, occurs when the wave direction is *exactly* opposite to the breakwater orientation, so that we have *exactly*  $\theta_0 = \pi$ . In this case the wave train shadow line and the breakwater are each other's extension in Figure 3. Because it is intrinsically unclear in this case from which side of the breakwater the incident wave is coming, the case  $\theta_0 = \pi$  forms singularity which may just produce different results for the Penney and Price method and the Goda method.

The functions within Appendices B and C do not only produce the resulting total wave field  $F$  as output variables; they also yield various quantities which together make up the final solution. The wave function  $F_i$  of the incident wave and the function  $F_r$  of the reflected wave can be retrieved separately, after which the user may add them up using some reflection coefficient  $R$  between 0.0 and 1.0, see Equation (4). (The standard solution  $F$  which is computed by these Matlab functions supposes the default value of  $R = 1$ .) Furthermore, the functions yield the quantities  $G_i$  and  $G_r$ , which are the Fresnel integral factors for the incident and reflected wave, and the quantities  $H_i$  and  $H_r$ , which are undisturbed wave fields for the incident and reflected wave. These quantities  $G$  and  $H$  are in fact equal to the first and second factor in each line of Equation (5). This means that we can interpret the Fresnel integral factor  $G$  as a *correction factor* on the original undisturbed wave  $H$ . Hence we can write:

$$F = F_i + R \cdot F_r = G_i H_i + R \cdot G_r H_r \quad (19)$$

If the breakwater head is chosen to be located very far away from the origin of the computational domain (it may lay even outside it!), so that the breakwater is virtually absent, we can derive from Equation (8) that  $\sigma_i \rightarrow \infty$  and  $\sigma_r \rightarrow -\infty$ . For those limits, we find that  $f(\sigma_i) \rightarrow \frac{1}{2}$  and  $f(\sigma_r) \rightarrow -\frac{1}{2}$ , so that  $G_i \rightarrow 1$  and  $G_r \rightarrow 0$ . Equation (19) then reduces to:

$$F = H_i \quad (20)$$

So in this limiting case, we find that the final solution is given by the undisturbed incident wave.

When the user wishes to perform some experiments with the Sommerfeld Matlab scripts in Appendix B and C, he/she will soon find out that it is vital that the polar coordinate angle  $\theta$  is in the range  $0 \leq \theta \leq 2\pi$ . The reason for this is that this choice locates the discontinuity in  $\theta$  exactly at



the location of the breakwater, which is located at  $\theta = 0$  by definition. Obviously, the breakwater forms the only discontinuity within an otherwise continuous physical problem; the discontinuity in  $\theta$  should reflect this fact. (If  $\theta$  would be chosen within the region  $-\pi \leq \theta \leq \pi$  instead, for example, then the discontinuity in  $\theta$  would be located in the direction opposite to the breakwater's location, which would produce incorrect results.)

## 4 SECOND STEP: APPROXIMATE SOLUTION FOR BREAKWATER GAP

### 4.1 Introduction: a re-construction strategy

The Sommerfeld solution is exact for the case of a single semi-infinite breakwater which extends to infinity from the origin along the positive x-axis. It is only approximately correct for the case of two breakwaters aligned with each other, with a gap in between that is at least larger than one wavelength. Our second step is to use the semi-infinite breakwater solution to construct an approximate solution for the breakwater gap case. This step is *not* described explicitly by Goda et al. (1978), although it has been a vital step for them to construct the Goda diffraction diagrams. Neither has it been described explicitly within any other literature that has been found so far, although some local analytical approximations have been given by Penney and Price (1952).

On order to re-construct this crucial step, we will first examine how the various components of the original Sommerfeld solution satisfy the Helmholtz Equation (1) and the boundary condition given by Equation (2). Subsequently, we will construct a solution for the breakwater gap problem and check to what extent this approximate solution satisfies the associated boundary conditions.

### 4.2 Satisfaction of Helmholtz Equation and boundary conditions

The Sommerfeld solution is usually given in polar coordinates, as can be seen from Equations (5) to (9). Therefore it is convenient to re-write the Helmholtz Equation (1) and boundary condition Equation (2) in terms of polar coordinates:

$$\nabla^2 F + k^2 F = \frac{\partial^2 F}{\partial r^2} + \frac{1}{r} \frac{\partial F}{\partial r} + \frac{1}{r^2} \frac{\partial^2 F}{\partial \theta^2} + k^2 F = 0, \text{ and} \quad (21)$$

$$\frac{\partial F}{\partial \theta} = 0 \quad \text{at} \quad \theta = 0 \quad \text{or} \quad \theta = 2\pi \quad (22)$$

Next, the Sommerfeld solution  $F = F_i + R \cdot F_r$  can be substituted into (21) and (22) in order to check whether these conditions are satisfied. We will consistently use the Penney and Price (1952) version of the Sommerfeld solution throughout from here.

Because the solutions for the incident wave and the reflected wave are quite similar except for some minus signs, we will define a *generalized* wave solution here which can be made either the incident wave or the reflected wave by setting some coefficients at a value +1 or -1. We define:

$$F = GH, \text{ with } G = \left\{ \frac{1}{2} + f(\sigma) \right\} \text{ and } H = \exp \{ -ikr \cos(\theta - s\theta_0) \}, \quad (23)$$

where the parameter  $s$  can be equal to +1 for the incident wave or -1 for the reflected wave. In this way we can obtain both equations within Equation (5). The formulation for  $f(\sigma)$  remains Equation (6), and the generalized formula for the integration boundary  $\sigma$  is given by:

$$\sigma = 2ps \sqrt{\frac{kr}{\pi}} \sin \left( \frac{\theta - s\theta_0}{2} \right), \quad (24)$$

where  $s$  is equal to the parameter  $s$  defined by Equation (23), and  $p$  is equal to the correction factor defined in Equation (8) (which means that is is either +1 or -1 as well):

$$p = \text{sign}\left\{\cos\left(\frac{\theta_0}{2}\right)\right\} \quad (25)$$

For use later on, we derive some practical expressions for partial derivatives:

$$\begin{aligned} \frac{\partial f(\sigma)}{\partial \sigma} &= \frac{1+i}{2} \exp\left\{-i\frac{1}{2}\pi\sigma^2\right\} = \frac{1+i}{2} \exp\left\{-i\frac{1}{2}\pi\left(4p^2s^2\frac{kr}{\pi}\sin^2\left(\frac{\theta-s\theta_0}{2}\right)\right)\right\} \\ &= \frac{1+i}{2} \exp\{-ikr(1-\cos(\theta-s\theta_0))\} \end{aligned} \quad (26)$$

$$\frac{\partial \sigma}{\partial r} = ps\sqrt{\frac{k}{\pi r}} \sin\left(\frac{\theta-s\theta_0}{2}\right) = \frac{\sigma}{2r} \quad (27)$$

$$\frac{\partial^2 \sigma}{\partial r^2} = -\frac{1}{2}ps\sqrt{\frac{k}{\pi r^3}} \sin\left(\frac{\theta-s\theta_0}{2}\right) = -\frac{\sigma}{4r^2} \quad (28)$$

$$\frac{\partial \sigma}{\partial \theta} = ps\sqrt{\frac{kr}{\pi}} \cos\left(\frac{\theta-s\theta_0}{2}\right) \quad (29)$$

$$\frac{\partial^2 \sigma}{\partial \theta^2} = -\frac{1}{2}ps\sqrt{\frac{kr}{\pi}} \sin\left(\frac{\theta-s\theta_0}{2}\right) = -\frac{\sigma}{4} \quad (30)$$

For convenience we define the auxiliary variable  $\tau$ , which is equal to twice  $\partial\sigma/\partial\theta$ :

$$\tau = 2\frac{\partial \sigma}{\partial \theta} = 2ps\sqrt{\frac{kr}{\pi}} \cos\left(\frac{\theta-s\theta_0}{2}\right) \quad (30a)$$

Furthermore we note the following relation, which is brief and convenient:

$$H\frac{\partial f(\sigma)}{\partial \sigma} = \frac{1+i}{2} \exp\{-ikr(1-\cos(\theta-s\theta_0))\} \exp\{-ikr\cos(\theta-s\theta_0)\} = \frac{1+i}{2} \exp\{-ikr\} \quad (31)$$

Now, we will derive various partial derivatives of the Sommerfeld solution (23). We have:

$$\frac{\partial F}{\partial r} = \frac{\partial G}{\partial r}H + G\frac{\partial H}{\partial r} = \frac{1+i}{2} \exp\{-ikr\} \frac{\sigma}{2r} + GH\{-ik\cos(\theta-s\theta_0)\}, \quad (32)$$

and therefore also:

$$\frac{1}{r}\frac{\partial F}{\partial r} = \underbrace{\frac{1+i}{2} \exp\{-ikr\} \frac{\sigma}{2r^2}}_{(A)} + \underbrace{\frac{GH}{r^2} \{-ikr\cos(\theta-s\theta_0)\}}_{(B)} \quad (33)$$

We will need the terms (A) and (B) later on. Taking another derivative from (32), we find:

$$\frac{\partial^2 F}{\partial r^2} = \underbrace{\frac{1+i}{2} \exp\{-ikr\} \frac{\sigma}{2r} \{-ik\}}_{(C)} + \underbrace{\frac{1+i}{2} \exp\{-ikr\} \frac{-\sigma}{4r^2}}_{(D)} + \underbrace{\frac{1+i}{2} \exp\{-ikr\} \frac{\sigma}{2r} \{-ik \cos(\theta - s\theta_0)\}}_{(E)} + \underbrace{GH \{-ik \cos(\theta - s\theta_0)\}^2}_{(F)} \quad (34)$$

Again, we will need the terms (C), (D), (E) and (F) later on. Furthermore, we have:

$$\frac{\partial F}{\partial \theta} = \frac{\partial G}{\partial \theta} H + G \frac{\partial H}{\partial \theta} = \frac{1+i}{2} \exp\{-ikr\} \frac{\tau}{2} + GH \{ikr \sin(\theta - s\theta_0)\} \quad (35)$$

This equation will become important when it comes to checking whether the Sommerfeld solution satisfies the boundary condition given by Equation (22). We will return later on to that issue.

Taking another derivative from (35), we find:

$$\frac{\partial^2 F}{\partial \theta^2} = \frac{1+i}{2} \exp\{-ikr\} \left\{ \frac{-\sigma}{4} \right\} + GH \{ikr \cos(\theta - s\theta_0)\} + \frac{1+i}{2} \exp\{-ikr\} \frac{\tau}{2} \{ikr \sin(\theta - s\theta_0)\} + GH \{ikr \sin(\theta - s\theta_0)\}^2 \quad (36)$$

and therefore also:

$$\frac{1}{r^2} \frac{\partial^2 F}{\partial \theta^2} = \underbrace{\frac{1+i}{2} \exp\{-ikr\} \left\{ -\frac{\sigma}{4r^2} \right\}}_{(G)} + \underbrace{\frac{GH}{r^2} \{ikr \cos(\theta - s\theta_0)\}}_{(H)} + \underbrace{\frac{1+i}{2} \exp\{-ikr\} \frac{\tau}{2r^2} \{ikr \sin(\theta - s\theta_0)\}}_{(I)} + \underbrace{\frac{GH}{r^2} \{ikr \sin(\theta - s\theta_0)\}^2}_{(J)} \quad (37)$$

Combining Equations (33), (34) and (37), we have sufficient information to construct the Laplacian operator within the Helmholtz Equation (21). This yields:

$$\begin{aligned}
& \frac{\partial^2 F}{\partial r^2} + \frac{1}{r} \frac{\partial F}{\partial r} + \frac{1}{r^2} \frac{\partial^2 F}{\partial \theta^2} = \\
& \frac{1+i}{2} \exp\{-ikr\} \left[ \underbrace{\frac{\sigma}{2r^2}}_{(A)} + \underbrace{\left(-\frac{\sigma}{4r^2}\right)}_{(G)} + \underbrace{\left(-\frac{\sigma}{4r^2}\right)}_{(D)} \right] + \\
& \frac{1+i}{2} \exp\{-ikr\} \left[ \underbrace{\frac{\sigma}{2r^2} \{-ikr\}}_{(C)} + \underbrace{\frac{\sigma}{2r^2} \{-ikr \cos(\theta - s\theta_0)\}}_{(E)} + \underbrace{\frac{\tau}{2r^2} \{ikr \sin(\theta - s\theta_0)\}}_{(I)} \right] + \quad (38) \\
& \frac{GH}{r^2} \left[ \underbrace{\{-ikr \cos(\theta - s\theta_0)\}^2}_{(F)} + \underbrace{\{ikr \sin(\theta - s\theta_0)\}^2}_{(J)} \right] + \\
& \frac{GH}{r^2} \left[ \underbrace{\{-ikr \cos(\theta - s\theta_0)\}}_{(B)} + \underbrace{\{ikr \cos(\theta - s\theta_0)\}}_{(H)} \right]
\end{aligned}$$

In (38), the 10 terms (A) to (J) have been grouped in a convenient way, where common factors have been placed outside the square brackets.

It can be readily seen that the terms (A), (G) and (D) cancel each other out, and also the terms (B) and (H) vanish. In order to analyze the terms (C), (E) and (I), we need to substitute the expressions for  $\sigma$  and  $\tau$  which are given in Equations (24) and (30a). This yields:

$$\begin{aligned}
(C) + (E) + (I) &= \frac{1+i}{2} \exp\{-ikr\} \frac{1}{2r^2} 2ps \sqrt{\frac{kr}{\pi}} \{ikr\} \cdot \\
& \left[ \underbrace{\sin\left(\frac{\theta - s\theta_0}{2}\right) \{-1\}}_{(C)} + \underbrace{\sin\left(\frac{\theta - s\theta_0}{2}\right) \{-\cos(\theta - s\theta_0)\}}_{(E)} + \underbrace{\cos\left(\frac{\theta - s\theta_0}{2}\right) \{\sin(\theta - s\theta_0)\}}_{(I)} \right] \quad (39)
\end{aligned}$$

Using the general goniometric rule  $\sin(\alpha - \beta) = \sin \alpha \cos \beta - \cos \alpha \sin \beta$  to unify term (E) and (I), we will find that the sum (C)+(E)+(I) is exactly zero. This means that *all* terms including the factor  $\frac{1}{2}(1+i)\exp\{-ikr\}$  vanish from Equation (38). Actually only the terms (F) and (J) remain, which can be simplified too. Using the well-known rule  $\sin^2 \alpha + \cos^2 \alpha = 1$ , we find:

$$\frac{\partial^2 F}{\partial r^2} + \frac{1}{r} \frac{\partial F}{\partial r} + \frac{1}{r^2} \frac{\partial^2 F}{\partial \theta^2} = \frac{GH}{r^2} \{-ikr\}^2 = -\frac{GH}{r^2} k^2 r^2 = -k^2 F, \quad (40)$$

from which it is finally found that:

$$\frac{\partial^2 F}{\partial r^2} + \frac{1}{r} \frac{\partial F}{\partial r} + \frac{1}{r^2} \frac{\partial^2 F}{\partial \theta^2} + k^2 F = \nabla^2 F + k^2 F = 0 \quad (41)$$

Now, we did not only prove that the Sommerfeld solution satisfies the Helmholtz Equation (21), something that we already expected. We have also proven that *each* incident or reflected wave

separately satisfies the Helmholtz problem already (!). This is an important and convenient result. It implies that we may superimpose various incident and reflected waves, dependent on the breakwater geometry, *without violating the Helmholtz Equation*.

The latter, however, is not true for the satisfaction of the boundary condition (22). From Equation (35), we see that there is no way in which a single incident or reflected wave could satisfy the impermeability condition  $\partial F / \partial \theta = 0$  at  $\theta = 0$  or  $\theta = 2\pi$ . It is only the *combination* of an incident wave and its associated *fully* reflected wave that satisfies (22):

$$\frac{\partial F}{\partial \theta} = \frac{\partial F_i}{\partial \theta} + \frac{\partial F_r}{\partial \theta} = \frac{1+i}{2} \exp\{-ikr\} \frac{\tau_i}{2} + G_i H_i \{ikr \sin(\theta - \theta_0)\} + \frac{1+i}{2} \exp\{-ikr\} \frac{\tau_r}{2} + G_r H_r \{ikr \sin(\theta + \theta_0)\} \quad (42)$$

This can be seen as follows. We first examine the boundary condition  $\theta = 0$ . We compute the corresponding values of the integration boundary parameters  $\sigma_i$  and  $\sigma_r$ :

$$\sigma_i = 2p \sqrt{\frac{kr}{\pi}} \sin\left(\frac{-\theta_0}{2}\right) \text{ and } \sigma_r = -2p \sqrt{\frac{kr}{\pi}} \sin\left(\frac{\theta_0}{2}\right) \quad (43)$$

Because  $\sin(-\alpha) = -\sin \alpha$ , we see that  $\sigma_i = \sigma_r = \sigma$ ; this also implies that  $G_i = G_r = G$ , because of  $f(\sigma_i) = f(\sigma_r) = f(\sigma)$ . Furthermore, we have:

$$H_i = \exp\{-ikr \cos(-\theta_0)\} \text{ and } H_r = \exp\{-ikr \cos(\theta_0)\} \quad (44)$$

Because  $\cos(-\alpha) = \cos \alpha$ , we find  $H_i = H_r = H$ . Now, we substitute the values of  $\tau_i$  and  $\tau_r$  into (42), where  $s = 1$  for the incident wave and  $s = -1$  for the reflected wave. This yields:

$$\begin{aligned} \frac{\partial F}{\partial \theta} &= \frac{1+i}{2} \exp\{-ikr\} p \sqrt{\frac{kr}{\pi}} \cos\left(\frac{-\theta_0}{2}\right) + GH \{ikr \sin(-\theta_0)\} + \\ &- \frac{1+i}{2} \exp\{-ikr\} p \sqrt{\frac{kr}{\pi}} \cos\left(\frac{\theta_0}{2}\right) + GH \{ikr \sin(\theta_0)\} = \\ &\frac{1+i}{2} \exp\{-ikr\} p \sqrt{\frac{kr}{\pi}} \left[ \cos\left(\frac{-\theta_0}{2}\right) - \cos\left(\frac{\theta_0}{2}\right) \right] + GH \{ikr\} [\sin(-\theta_0) + \sin(\theta_0)] = 0 \end{aligned} \quad (45)$$

So boundary condition (22) is satisfied for  $\theta = 0$ . Now we turn to the case  $\theta = 2\pi$ . First we compute the corresponding values of the integration boundary parameters  $\sigma_i$  and  $\sigma_r$ :

$$\sigma_i = 2p \sqrt{\frac{kr}{\pi}} \sin\left(\pi - \frac{\theta_0}{2}\right) \text{ and } \sigma_r = -2p \sqrt{\frac{kr}{\pi}} \sin\left(\pi + \frac{\theta_0}{2}\right) \quad (46)$$

Because  $\sin(\pi - \alpha) = \sin(\pi + \alpha)$ , we see that  $\sigma_i = \sigma_r = \sigma$ ; this also implies that  $G_i = G_r = G$ , because of  $f(\sigma_i) = f(\sigma_r) = f(\sigma)$ . Furthermore, we have:

$$H_i = \exp\{-ikr \cos(2\pi - \theta_0)\} \text{ and } H_r = \exp\{-ikr \cos(2\pi + \theta_0)\} \quad (47)$$

Because  $\cos(2\pi - \alpha) = \cos(2\pi + \alpha)$ , we find  $H_i = H_r = H$ . Now, we substitute the values of  $\tau_i$  and  $\tau_r$  into (42), where  $s = 1$  for the incident wave and  $s = -1$  for the reflected wave. This yields:

$$\begin{aligned}
 \frac{\partial F}{\partial \theta} &= \frac{1+i}{2} \exp\{-ikr\} p \sqrt{\frac{kr}{\pi}} \cos\left(\pi - \frac{\theta_0}{2}\right) + GH\{ikr \sin(2\pi - \theta_0)\} + \\
 &- \frac{1+i}{2} \exp\{-ikr\} p \sqrt{\frac{kr}{\pi}} \cos\left(\pi + \frac{\theta_0}{2}\right) + GH\{ikr \sin(2\pi + \theta_0)\} = \\
 &\frac{1+i}{2} \exp\{-ikr\} p \sqrt{\frac{kr}{\pi}} \left[ \cos\left(\frac{-\theta_0}{2}\right) - \cos\left(\frac{\theta_0}{2}\right) \right] + GH\{ikr\} [\sin(-\theta_0) + \sin(\theta_0)] = 0
 \end{aligned} \tag{48}$$

So boundary condition (22) is satisfied for the case  $\theta = 2\pi$  as well.

After all, we find that the mathematical boundary condition (22) at the breakwater is satisfied only for the case of a reflection coefficient  $R = 1$ . This boundary condition is thus far less flexible than the governing Helmholtz wave equation (21). No exact boundary condition can be derived for partial reflection or no-reflection conditions, although modified solutions exist (see e.g. Daemrich et al., 1978).

It is however pointed out by Penney and Price (1952) that another theoretical boundary condition can be derived too, namely:

$$F = 0 \quad \text{at } \theta = 0 \quad \text{or } \theta = 2\pi \tag{49}$$

This boundary condition is called the “cushion” boundary condition, which represents a pressure along the breakwater which remains constant and equal to the hydrostatic pressure at all depths. Although such type of breakwaters are probably of no practical importance, Penney and Price (1952) note that *“it is of interest to remark that some examples of [them] were designed, constructed and tested during the [Second World] War, and were found to behave in the manner predicted by theoretical considerations”*. On a theoretical level, the “cushion”-type of breakwater will play a small role in this chapter.

From the Equations (43) to (48), we have seen that  $G_i = G_r = G$  and  $H_i = H_r = H$  for both  $\theta = 0$  and  $\theta = 2\pi$ . This means that  $F_i = G_i H_i = G_r H_r = F_r$  on the breakwater boundary. Because we have defined the total wave solution as  $F = F_i + R \cdot F_r$ , we find that we can mathematically enforce  $F = 0$  on  $\theta = 0$  and  $\theta = 2\pi$  if we set the reflection coefficient to  $R = -1$ . So we find:

- The boundary condition (22) for rigid breakwaters is satisfied by  $R = 1$ ;
- The boundary condition (49) for “cushion”-type breakwaters is satisfied by  $R = -1$ .

Finally, an important remark is made here about *phase shifts* of the Sommerfeld solution. To the basic generalized Sommerfeld solution given by Equation (23) we may add *any* phase difference  $\phi$  which is not a function of  $r$  and  $\theta$ , without violating the satisfaction of the Helmholtz Equation (21) and the boundary conditions (22) and (49). So we can write:

$$F = GHI, \text{ with } I = \exp\{i\phi\}, \tag{50}$$

where  $G$  and  $H$  are given by Equation (23). Because the complex number  $I = \exp\{i\phi\}$  is in fact a constant (invariant) scalar quantity, while the equations (21), (22) and (49) are linear in  $F$ , we find that  $\phi$  may have any real value in radians without destroying the validity of the Sommerfeld solution (!). This enables us to give Sommerfeld solutions a “shift” in two-dimensional space if convenient. The latter will be important for the construction of the approximate solution for the breakwater gap.

### 4.3 “Right half” and “left half” of the breakwater gap solution

As a next step toward the approximate breakwater gap solution, we will now construct the exact Sommerfeld solutions for each of the two breakwaters separately; this will be done using the terminology from Section 3.6.

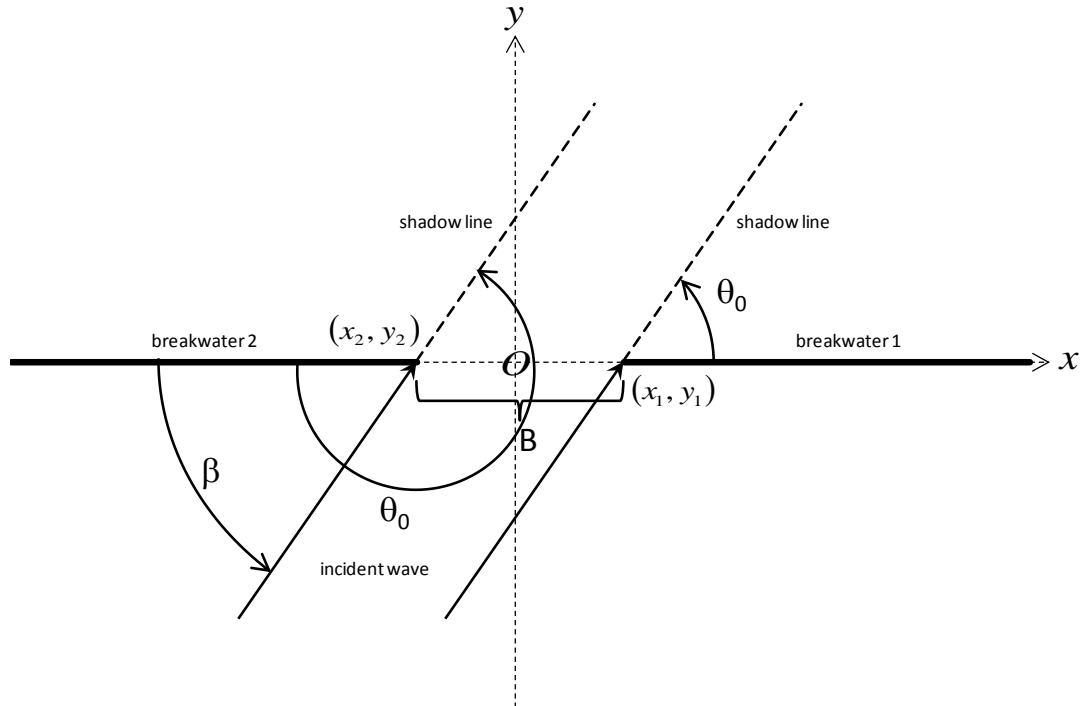


Figure 4: Geometry of the wave diffraction problem with a gap between two breakwaters.

We define two breakwaters aligned with the  $x$ -axis, with a gap of width  $B$  in between them (see Figure 4). Breakwater 1 coincides with the positive  $x$ -axis from the point  $x = \frac{1}{2}B$  toward infinity, and breakwater 2 coincides with the negative  $x$ -axis from the point  $x = -\frac{1}{2}B$  toward minus infinity. The coordinates of the breakwater heads are respectively:  $(x_1, y_1) = (\frac{1}{2}B, 0)$  and  $(x_2, y_2) = (-\frac{1}{2}B, 0)$ . The global wave direction angle is  $\beta$ . According to the solution given in Section 3.6, we have  $\delta = 0$  and local wave direction angle  $\theta_0 = \beta$  for breakwater 1, and on the contrary  $\delta = \pi$  and local wave direction angle  $\theta_0 = \beta - \pi$  for breakwater 2.

For both breakwaters we can now define a local coordinate system, which is either Cartesian  $(\bar{x}_1, \bar{y}_1)$  and  $(\bar{x}_2, \bar{y}_2)$  or polar  $(r_1, \theta_1)$  and  $(r_2, \theta_2)$  according to Equations (17) and (18). Our objective is now to express the Sommerfeld solutions for both breakwaters in terms of the local polar coordinate system and the global wave direction angle.

For breakwater 1, the expressions for  $\sigma_i$  and  $\sigma_r$  are:

$$\sigma_{i1} = 2\sqrt{\frac{kr_1}{\pi}} \sin\left(\frac{\theta_1 - \beta}{2}\right) \text{sign}\left\{\cos\left(\frac{\beta}{2}\right)\right\} \quad (51)$$



$$\sigma_{r1} = -2\sqrt{\frac{kr_1}{\pi}} \sin\left(\frac{\theta_1 + \beta}{2}\right) \text{sign}\left\{\cos\left(\frac{\beta}{2}\right)\right\} \quad (52)$$

For breakwater 2, the expressions for  $\sigma_i$  and  $\sigma_r$  are:

$$\sigma_{i2} = 2\sqrt{\frac{kr_2}{\pi}} \sin\left(\frac{\theta_2 - \beta + \pi}{2}\right) \text{sign}\left\{\cos\left(\frac{\beta - \pi}{2}\right)\right\} \quad (53)$$

$$\sigma_{r2} = -2\sqrt{\frac{kr_2}{\pi}} \sin\left(\frac{\theta_2 + \beta - \pi}{2}\right) \text{sign}\left\{\cos\left(\frac{\beta - \pi}{2}\right)\right\} \quad (54)$$

with  $0 \leq \theta_1, \theta_2 \leq 2\pi$  (!). The solutions that we are looking for are now given by:

$$F_1 = G_{i1}H_{i1} + RG_{r1}H_{r1} = \left\{\frac{1}{2} + f(\sigma_{i1})\right\} \exp\{-ikr_1 \cos(\theta_1 - \beta)\} + R\left\{\frac{1}{2} + f(\sigma_{r1})\right\} \exp\{-ikr_1 \cos(\theta_1 + \beta)\} \quad (55)$$

for breakwater 1 (right side), and

$$F_2 = G_{i2}H_{i2} + RG_{r2}H_{r2} = \left\{\frac{1}{2} + f(\sigma_{i2})\right\} \exp\{-ikr_2 \cos(\theta_2 - \beta + \pi)\} + R\left\{\frac{1}{2} + f(\sigma_{r2})\right\} \exp\{-ikr_2 \cos(\theta_2 + \beta - \pi)\} \quad (56)$$

for breakwater 2 (left side).

We observe that a phase difference exists between the undisturbed incident wave fields  $H_{i1}$  and  $H_{i2}$ . In order to combine the two solutions given in (55) and (56), we will have to eliminate this phase difference in a way that is consistent with the Helmholtz problem and the governing boundary conditions.

#### 4.4 Solution assembly: approximate solution of the breakwater gap problem

Each of the solutions (55) and (56) satisfies the Helmholtz problem (21) separately; moreover, each of these solutions satisfies as such boundary condition (22) for  $R = 1$ , or boundary condition (49) for  $R = -1$ . In order to combine the two single-breakwater problems into one consistent two-breakwater solution, we must combine their respective integration boundaries  $\sigma$  (for the incident wave) into one single Fresnel integral.

For a single breakwater (numbered with number 1), the Fresnel integral factor within the Sommerfeld solution reads (for the incident wave):

$$G = \left\{\frac{1}{2} + f(\sigma_{i1})\right\} = \frac{1}{2} + \frac{1+i}{2} \int_0^{\sigma_{i1}} \exp\{-i\frac{1}{2}\pi u^2\} du \quad (57)$$

In fact, the term  $\frac{1}{2}$  represents the value of the Fresnel integral between zero and minus infinity. So alternatively we could write:

$$G = \frac{1+i}{2} \int_{-\infty}^{\sigma_{i1}} \exp\left\{-i\frac{1}{2}\pi t^2\right\} du \quad (58)$$

Basically, this is a continuous summation of point sources along a line (according to Huygens' principle of wave propagation). Because in the basic Penney and Price (1952) solution a breakwater is present at the right side of the domain only, the left integration boundary is equal to minus infinity. If there is however a left breakwater indeed, we should replace  $-\infty$  by the corresponding integration boundary of the left breakwater. Then we obtain:

$$G = \frac{1+i}{2} \int_{\sigma_{i2}}^{\sigma_{i1}} \exp\left\{-i\frac{1}{2}\pi t^2\right\} du = f(\sigma_{i1}) - f(\sigma_{i2}) \quad (59)$$

Plotted in the complex plane, this integral represents a straight vector from the point  $f(\sigma_{i2})$  toward the point  $f(\sigma_{i1})$  on the Cornu spiral. This vector is defined for each location in the 2D spatial domain. In fact,  $\sigma_{i1}$  and  $\sigma_{i2}$  are both defined for each location in the 2D spatial domain; therefore it is legal to use them both as integration limits within the same Fresnel integral. However, because  $\sigma_{i1}$  and  $\sigma_{i2}$  have been defined within two different polar coordinate systems with a  $180^\circ$  difference in orientation (see Equations (51) and (53)), we must *swap a minus sign* in order to superimpose  $f(\sigma_{i1})$  and  $f(\sigma_{i2})$  in a correct way. Therefore we must replace Equation (59) by:

$$G = f(\sigma_{i1}) + f(\sigma_{i2}) \quad (60)$$

In terms of the  $G$ -factors that we already know, we may alternatively write:

$$G = \left\{\frac{1}{2} + f(\sigma_{i1})\right\} + \left\{\frac{1}{2} + f(\sigma_{i2})\right\} - 1 = G_{i1} + G_{i2} - 1 \quad (61)$$

We have shown now that the Fresnel integrals belonging to the single-breakwater solutions (55) and (56) may be superimposed for the incident wave. However, this is obviously only allowed if  $H_{i1} = H_{i2}$ , yet we have already observed that there is a phase difference between  $H_{i1} = \exp\{-ikr_1 \cos(\theta_1 - \beta)\}$  and  $H_{i2} = \exp\{-ikr_2 \cos(\theta_2 - \beta + \pi)\}$ . This is due to the fact that  $H_{i1}$  has a phase equal to zero in the origin of its associated polar coordinate system, which is the head of breakwater 1, whereas  $H_{i2}$  has a phase equal to zero at the head of breakwater 2. Fortunately, this phase difference is not a function of  $r$  and  $\theta$ , and therefore we may eliminate it by giving both solutions a constant phase shift. We will add a phase shift to both solutions in such a way that they will both yield a zero phase in the origin of the global Cartesian coordinate system, which is equal for both breakwaters, namely exactly half-way the breakwater gap:  $(x, y) = (0, 0)$ .

The phase difference  $\phi$  between any breakwater head and the point half-way the breakwater gap is equal to:

$$\phi = \frac{1}{2} kB \cos \beta \quad (62)$$

Based on this value we define two complex factors  $I_1 = \exp\{-i\phi\}$  and  $I_2 = \exp\{i\phi\}$ , which are added to equation (55) and (56) in order to synchronize both solutions. It is important that, for each solution, both the incident and the reflected wave are multiplied by the same phase shifting factor. Then the solutions (55) and (56) become:

$$F_1 = G_{i1}H_{i1}I_1 + RG_{r1}H_{r1}I_1 \quad (63)$$

for breakwater 1 (right side), and

$$F_2 = G_{i2}H_{i2}I_2 + RG_{r2}H_{r2}I_2 \quad (64)$$

for breakwater 2 (left side).

The incident waves within (63) and (64) can be combined using Equation (61), whereas the reflected waves do not need further adaptation. Then we obtain the following **final solution for the breakwater gap case with monochromatic waves**:

$$F = (G_{i1} + G_{i2} - 1)H_{i0} + R_1G_{r1}H_{r1}I_1 + R_2G_{r2}H_{r2}I_2, \quad (65)$$

in which the two reflection coefficients  $R_1$  and  $R_2$  may have a different value for each breakwater. Furthermore, we have  $H_{i0} = H_{i1}I_1 = H_{i2}I_2$  and also  $H_{r0} = H_{r1}I_1 = H_{r2}I_2$  thanks to the applied phase shifting. We will check this:

$$\begin{aligned} H_{i1}I_1 &= \exp\{-ikr_1 \cos(\theta_1 - \beta)\} \exp\{-i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{-ikr_1 \cos \theta_1 \cos \beta - ikr_1 \sin \theta_1 \sin \beta - i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{-ik\bar{x}_1 \cos \beta - ik\bar{y}_1 \sin \beta - i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{-ik(x - \frac{1}{2}B) \cos \beta - ik y \sin \beta - i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{-ikx \cos \beta - ik y \sin \beta\} = H_{i0} \end{aligned} \quad (66)$$

$$\begin{aligned} H_{i2}I_2 &= \exp\{-ikr_2 \cos(\theta_2 - \beta + \pi)\} \exp\{i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{ikr_2 \cos(\theta_2 - \beta)\} \exp\{i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{ikr_2 \cos \theta_2 \cos \beta + ikr_2 \sin \theta_2 \sin \beta + i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{ik\bar{x}_2 \cos \beta + ik\bar{y}_2 \sin \beta + i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{ik(-\frac{1}{2}B - x) \cos \beta + ik(-y) \sin \beta + i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{-ikx \cos \beta - ik y \sin \beta\} = H_{i0} \end{aligned} \quad (67)$$

for the incident waves, and:

$$\begin{aligned} H_{r1}I_1 &= \exp\{-ikr_1 \cos(\theta_1 + \beta)\} \exp\{-i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{-ikr_1 \cos \theta_1 \cos \beta + ikr_1 \sin \theta_1 \sin \beta - i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{-ik\bar{x}_1 \cos \beta + ik\bar{y}_1 \sin \beta - i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{-ik(x - \frac{1}{2}B) \cos \beta + ik y \sin \beta - i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{-ikx \cos \beta + ik y \sin \beta\} = H_{r0} \end{aligned} \quad (68)$$

$$\begin{aligned} H_{r2}I_2 &= \exp\{-ikr_2 \cos(\theta_2 + \beta - \pi)\} \exp\{i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{ikr_2 \cos(\theta_2 + \beta)\} \exp\{i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{ikr_2 \cos \theta_2 \cos \beta - ikr_2 \sin \theta_2 \sin \beta + i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{ik\bar{x}_2 \cos \beta - ik\bar{y}_2 \sin \beta + i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{ik(-\frac{1}{2}B - x) \cos \beta - ik(-y) \sin \beta + i\frac{1}{2}kB \cos \beta\} \\ &= \exp\{-ikx \cos \beta + ik y \sin \beta\} = H_{r0} \end{aligned} \quad (69)$$

for the reflected waves. This shows that the applied phase shifting works satisfactory.

#### 4.5 Assembled solution: approximate satisfaction of boundary conditions

In this section, we will show that the boundary conditions for the breakwater gap case are almost satisfied, yet not fully. This means that the final solution (65) is approximate, which is mainly due to scatter caused by the left reflected wave on the boundary of the right breakwater, and vice versa.

First, we note that the Helmholtz Equation (21) is satisfied by (65) under all circumstances. We may rewrite (65) as:

$$F = \left( \underbrace{G_{i1}H_{i1}I_1}_{(A)} + \underbrace{R_1G_{r1}H_{r1}I_1}_{(B)} \right) + \left( \underbrace{G_{i2}H_{i2}I_2}_{(C)} + \underbrace{R_2G_{r2}H_{r2}I_2}_{(D)} \right) - \underbrace{H_{i0}}_{(E)}, \quad (70)$$

in which term (A) and (B) represent the solution for a single right breakwater, term (C) and (D) represent the solution for a single left breakwater, and term (E) represents an undisturbed wave field which needs to be subtracted from the previous summation to obtain a correct superposition of both solutions. Based on the results given by Equations (23) to (41) (see Section 4.2), we state that each of the terms (A), (B), (C) and (D) satisfies the Helmholtz Equation (21) separately; furthermore, the undisturbed wave field  $H_{i0}$  within term (E) satisfies Equation (21) trivially. Therefore we conclude that *each of these five terms readily satisfies the Helmholtz Equation (21)*; hence, their summation satisfies (21) as well.

The next task is to examine to what extent the boundary condition (22) is satisfied for the two-breakwater case with a gap in between. Because the equations (65) and (70) are defined on more than one coordinate system (in fact there are three, namely the systems  $(r_1, \theta_1)$ ,  $(r_2, \theta_2)$  and  $(x, y)$ ), we cannot use the boundary condition in the form given by (22). Instead, we must use:

$$\frac{\partial F}{\partial y} = 0 \quad \text{at } (\theta_1 = 0 \vee \theta_1 = 2\pi) \wedge \theta_2 = \pi \wedge y = 0 \quad (\text{breakwater 1), and} \quad (71)$$

$$\frac{\partial F}{\partial y} = 0 \quad \text{at } (\theta_2 = 0 \vee \theta_2 = 2\pi) \wedge \theta_1 = \pi \wedge y = 0 \quad (\text{breakwater 2}) \quad (72)$$

We will split up Equation (70) in three terms now:

$$F = F_1 + F_2 + F_3, \text{ with} \quad (73)$$

$$F_1(r_1, \theta_1) = G_{i1}H_{i1}I_1 + R_1G_{r1}H_{r1}I_1, \quad (74)$$

$$F_2(r_2, \theta_2) = G_{i2}H_{i2}I_2 + R_2G_{r2}H_{r2}I_2 \text{ and} \quad (75)$$

$$F_3(x, y) = -H_{i0} \quad (76)$$

In each of these three terms, only one coordinate system is involved. We now define the following transformations of partial derivatives:

$$\frac{\partial F_1}{\partial y_1} = \frac{\partial F_1}{\partial r_1} \sin \theta_1 + \frac{1}{r_1} \frac{\partial F_1}{\partial \theta_1} \cos \theta_1 \quad (77)$$

$$\frac{\partial F_2}{\partial \bar{y}_2} = \frac{\partial F_2}{\partial r_2} \sin \theta_2 + \frac{1}{r_2} \frac{\partial F_2}{\partial \theta_2} \cos \theta_2 \quad (78)$$

For breakwater 1, we have  $(\theta_1 = 0 \vee \theta_1 = 2\pi) \wedge \theta_2 = \pi$ ; for breakwater 2, we have  $(\theta_2 = 0 \vee \theta_2 = 2\pi) \wedge \theta_1 = \pi$ . In all of these cases, the terms with  $\sin \theta_1$  and  $\sin \theta_2$  vanish from (77) and (78), whereas the terms with  $\cos \theta_1$  and  $\cos \theta_2$  are either 1 or  $-1$ . If we furthermore substitute  $\partial \bar{y}_1 / \partial y = 1$  and  $\partial \bar{y}_2 / \partial y = -1$ , we get:

$$\frac{\partial F_1}{\partial y} = \frac{\partial F_1}{\partial \bar{y}_1} \frac{\partial \bar{y}_1}{\partial y} = \frac{1}{r_1} \frac{\partial F_1}{\partial \theta_1} \quad \text{and} \quad \frac{\partial F_2}{\partial y} = \frac{\partial F_2}{\partial \bar{y}_2} \frac{\partial \bar{y}_2}{\partial y} = -\frac{1}{r_2} \frac{\partial F_2}{\partial \theta_2} \quad (\text{at breakwater 1}) \quad (79)$$

$$\frac{\partial F_1}{\partial y} = \frac{\partial F_1}{\partial \bar{y}_1} \frac{\partial \bar{y}_1}{\partial y} = -\frac{1}{r_1} \frac{\partial F_1}{\partial \theta_1} \quad \text{and} \quad \frac{\partial F_2}{\partial y} = \frac{\partial F_2}{\partial \bar{y}_2} \frac{\partial \bar{y}_2}{\partial y} = \frac{1}{r_2} \frac{\partial F_2}{\partial \theta_2} \quad (\text{at breakwater 2}) \quad (80)$$

Now we can use the general expression for  $\partial F / \partial \theta$  in Equation (35) to derive the spatial derivatives that we are looking for.

We will first look at the boundary condition for breakwater 1. Taking the derivative of (74) yields:

$$\begin{aligned} \frac{\partial F_1}{\partial y} = & \frac{1}{r_1} \frac{\partial (G_{i1} H_{i1} I_1 + R_1 G_{r1} H_{r1} I_1)}{\partial \theta_1} = \frac{I_1}{2r_1} \frac{1+i}{2} \exp\{-ikr_1\} (\tau_{i1} + R_1 \tau_{r1}) + \\ & \frac{I_1}{r_1} G_{i1} H_{i1} \{ikr_1 \sin(\theta_1 - \beta)\} + \frac{R_1 I_1}{r_1} G_{r1} H_{r1} \{ikr_1 \sin(\theta_1 + \beta)\} \end{aligned} \quad (81)$$

From Equations (43) to (48), we know that  $\tau_{i1} = -\tau_{r1}$ ,  $G_{i1} = G_{r1}$  and also  $H_{i1} = H_{r1}$  for  $\theta_1 = 0$  as well as for  $\theta_1 = 2\pi$ . If we use these results, we obtain:

$$\frac{\partial F_1}{\partial y} = \left[ \frac{I_1}{2r_1} \frac{1+i}{2} \exp\{-ikr_1\} \tau_{i1} - G_{i1} H_{i1} I_1 ik \sin \beta \right] (1 - R_1) \quad (82)$$

We see that  $\partial F_1 / \partial y$  will be precisely zero if  $R_1 = 1$ , i.e. full reflection along breakwater 1 (not surprisingly, as this result would also be found for a single semi-infinite breakwater case).

Taking the derivative of (75) with respect to breakwater 1 yields:

$$\begin{aligned} \frac{\partial F_2}{\partial y} = & \frac{1}{r_2} \frac{\partial (G_{i2} H_{i2} I_2 + R_2 G_{r2} H_{r2} I_2)}{\partial \theta_2} = \frac{I_2}{2r_2} \frac{1+i}{2} \exp\{-ikr_2\} (\tau_{i2} + R_2 \tau_{r2}) + \\ & \frac{I_2}{r_2} G_{i2} H_{i2} \{ikr_2 \sin(\theta_2 - \beta + \pi)\} + \frac{R_2 I_2}{r_2} G_{r2} H_{r2} \{ikr_2 \sin(\theta_2 + \beta - \pi)\} \end{aligned} \quad (83)$$

For  $\theta_2 = \pi$  we have  $H_{i2} = H_{r2}$  and  $\tau_{i2} = \tau_{r2}$ , but  $G_{i2} = 1 - G_{r2}$  (!). If we use these results, we obtain:

$$\frac{\partial F_2}{\partial y} = \left[ \frac{I_2}{2r_2} \frac{1+i}{2} \exp\{-ikr_2\} \tau_{i2} - G_{i2} H_{i2} I_2 ik \sin \beta \right] (1 + R_2) + \underbrace{R_2 H_{i0} ik \sin \beta}_{(T)} \quad (84)$$

We see that most terms of  $\partial F_2 / \partial y$  will be precisely zero if  $R_2 = -1$ , i.e. negative reflection along breakwater 2. This would be the case if breakwater 2 would be of the ‘‘cushion type’’, see Section 4.2

and the boundary condition in Equation (49). The only non-zero term in (84) in that case is the last term, which is coined term (T) here.

Finally, taking the derivative of (76) with respect to breakwater 1 yields:

$$\frac{\partial F_3}{\partial y} = -\frac{\partial H_{i0}}{\partial y} = -\frac{\partial \exp\{-ikx \cos \beta - ikys \sin \beta\}}{\partial y} = H_{i0} ik \sin \beta \quad (85)$$

For  $R_2 = -1$ , this result plus term (T) from Equation (84) is exactly zero, so that both terms vanish. Now, if we take the sum of Equations (82), (84) and (85), we finally find the total derivative  $\partial F / \partial y$  along breakwater 1:

$$\begin{aligned} \frac{\partial F}{\partial y} = \frac{\partial F_1}{\partial y} + \frac{\partial F_2}{\partial y} + \frac{\partial F_3}{\partial y} = & \left[ \frac{I_1}{2r_1} \frac{1+i}{2} \exp\{-ikr_1\} \tau_{i1} - G_{i1} H_{i1} I_1 ik \sin \beta \right] (1 - R_1) \\ & + \left[ \frac{I_2}{2r_2} \frac{1+i}{2} \exp\{-ikr_2\} \tau_{i2} - G_{i2} H_{i2} I_2 ik \sin \beta \right] (1 + R_2) + (1 + R_2) H_{i0} ik \sin \beta \end{aligned} \quad (86)$$

This expression is exactly zero if  $R_1 = 1$  and  $R_2 = -1$ . In other words, the “rigid” boundary condition (22) for breakwater 1 can only be satisfied if breakwater 2 is of “cushion” type. Consequently, we must prove therefore that the “cushion” boundary condition for breakwater 2 is satisfied as well when we set the reflection coefficients to  $R_1 = 1$  and  $R_2 = -1$ . This means that we must prove the following boundary condition:

$$F = F_1 + F_2 + F_3 = 0 \quad \text{for } (\theta_2 = 0 \vee \theta_2 = 2\pi) \wedge \theta_1 = \pi \quad (\text{at breakwater 2}) \quad (87)$$

The proof is as follows. For  $\theta_1 = \pi$  we have  $H_{i1} = H_{r1}$  and  $\sigma_{i1} = -\sigma_{r1}$ , so that  $G_{i1} = 1 - G_{r1}$ . If we use these results, we find for  $F_1$ :

$$F_1 = G_{i1} H_{i1} I_1 + R_1 G_{r1} H_{r1} I_1 = G_{i1} H_{i1} I_1 (1 - R_1) + R_1 H_{i1} I_1 \quad (88)$$

Equivalently we find for  $F_2$ , with  $H_{i2} = H_{r2}$  and  $G_{i2} = G_{r2}$  (for  $\theta_2 = 0$  or  $\theta_2 = 2\pi$ ):

$$F_2 = G_{i2} H_{i2} I_2 + R_2 G_{r2} H_{r2} I_2 = (1 + R_2) G_{i2} H_{i2} I_2 \quad (89)$$

Finally,  $F_3$  is simply equal to:

$$F_3(x, y) = -H_{i0} = -H_{i1} I_1 \quad (90)$$

Summation of these three terms finally yields:

$$F = F_1 + F_2 + F_3 = G_{i1} H_{i1} I_1 (1 - R_1) + G_{i2} H_{i2} I_2 (1 + R_2) - H_{i1} I_1 (1 - R_1) \quad (91)$$

This expression is (again) exactly zero if  $R_1 = 1$  and  $R_2 = -1$  (!).

Now, we will turn to the boundary condition for breakwater 2. Taking the derivative of (74) yields:

$$\begin{aligned} \frac{\partial F_1}{\partial y} = & -\frac{1}{r_1} \frac{\partial (G_{i1} H_{i1} I_1 + R_1 G_{r1} H_{r1} I_1)}{\partial \theta_1} = -\frac{I_1}{2r_1} \frac{1+i}{2} \exp\{-ikr_1\} (\tau_{i1} + R_1 \tau_{r1}) - \\ & \frac{I_1}{r_1} G_{i1} H_{i1} \{ikr_1 \sin(\theta_1 - \beta)\} - \frac{R_1 I_1}{r_1} G_{r1} H_{r1} \{ikr_1 \sin(\theta_1 + \beta)\} \end{aligned} \quad (92)$$

For  $\theta_1 = \pi$  we have  $\tau_{i1} = \tau_{r1}$ ,  $H_{i1} = H_{r1}$  and also  $\sigma_{i1} = -\sigma_{r1}$ , so that  $G_{i1} = 1 - G_{r1}$ . If we use these results, we obtain:

$$\frac{\partial F_1}{\partial y} = \left[ -\frac{I_1}{2r_1} \frac{1+i}{2} \exp\{-ikr_1\} \tau_{i1} - G_{i1} H_{i1} I_1 ik \sin \beta \right] (1 + R_1) + \underbrace{R_1 H_{i0} ik \sin \beta}_{(T)} \quad (93)$$

Most terms of  $\partial F_1 / \partial y$  will be precisely zero if  $R_1 = -1$ , i.e. negative reflection along breakwater 1 (cushion-type breakwater). The only non-zero term in (93) in that case is the last term, which is coined term (T) again.

Taking the derivative of (75) with respect to breakwater 2 yields:

$$\begin{aligned} \frac{\partial F_2}{\partial y} = & -\frac{1}{r_2} \frac{\partial (G_{i2} H_{i2} I_2 + R_2 G_{r2} H_{r2} I_2)}{\partial \theta_2} = -\frac{I_2}{2r_2} \frac{1+i}{2} \exp\{-ikr_2\} (\tau_{i2} + R_2 \tau_{r2}) - \\ & \frac{I_2}{r_2} G_{i2} H_{i2} \{ikr_2 \sin(\theta_2 - \beta + \pi)\} - \frac{R_2 I_2}{r_2} G_{r2} H_{r2} \{ikr_2 \sin(\theta_2 + \beta - \pi)\} \end{aligned} \quad (94)$$

For  $\theta_2 = 0$  or  $\theta_2 = 2\pi$  we have  $\tau_{i2} = -\tau_{r2}$ ,  $H_{i2} = H_{r2}$  and also  $\sigma_{i2} = \sigma_{r2}$ , so that  $G_{i2} = G_{r2}$ . If we use these results, we obtain:

$$\frac{\partial F_2}{\partial y} = \left[ -\frac{I_2}{2r_2} \frac{1+i}{2} \exp\{-ikr_2\} \tau_{i2} - G_{i2} H_{i2} I_2 ik \sin \beta \right] (1 - R_2) \quad (95)$$

We see that  $\partial F_2 / \partial y$  will be precisely zero if  $R_2 = 1$ , i.e. full reflection along breakwater 2.

Finally, taking the derivative of (76) with respect to breakwater 2 yields:

$$\frac{\partial F_3}{\partial y} = -\frac{\partial H_{i0}}{\partial y} = -\frac{\partial \exp\{-ikx \cos \beta -iky \sin \beta\}}{\partial y} = H_{i0} ik \sin \beta \quad (96)$$

For  $R_1 = -1$ , this result plus term (T) from Equation (93) is exactly zero, so that both terms vanish. Now, if we take the sum of Equations (93), (95) and (96), we finally find the total derivative  $\partial F / \partial y$  along breakwater 2:

$$\begin{aligned} \frac{\partial F}{\partial y} = & \frac{\partial F_1}{\partial y} + \frac{\partial F_2}{\partial y} + \frac{\partial F_3}{\partial y} = \left[ -\frac{I_1}{2r_1} \frac{1+i}{2} \exp\{-ikr_1\} \tau_{i1} - G_{i1} H_{i1} I_1 ik \sin \beta \right] (1 + R_1) \\ & + \left[ -\frac{I_2}{2r_2} \frac{1+i}{2} \exp\{-ikr_2\} \tau_{i2} - G_{i2} H_{i2} I_2 ik \sin \beta \right] (1 - R_2) + (1 + R_1) H_{i0} ik \sin \beta \end{aligned} \quad (97)$$

This expression is exactly zero if  $R_1 = -1$  and  $R_2 = 1$ . In other words, the “rigid” boundary condition (22) for breakwater 2 can only be satisfied if breakwater 1 is of “cushion” type. Consequently, we must prove therefore that the “cushion” boundary condition for breakwater 1 is satisfied as well when we set the reflection coefficients to  $R_1 = -1$  and  $R_2 = 1$ . This means that we must prove the following boundary condition:

$$F = F_1 + F_2 + F_3 = 0 \quad \text{for } (\theta_1 = 0 \vee \theta_1 = 2\pi) \wedge \theta_2 = \pi \quad \text{(at breakwater 1)} \quad (98)$$

The proof is as follows. For  $\theta_1 = 0$  or  $\theta_1 = 2\pi$  we have  $H_{i1} = H_{r1}$  and  $\sigma_{i1} = \sigma_{r1}$ , so that  $G_{i1} = G_{r1}$ . If we use these results, we find for  $F_1$ :

$$F_1 = G_{i1}H_{i1}I_1 + R_1G_{r1}H_{r1}I_1 = G_{i1}H_{i1}I_1(1 + R_1) \quad (99)$$

Equivalently we find for  $F_2$ , with  $H_{i2} = H_{r2}$  and  $G_{i2} = 1 - G_{r2}$  (for  $\theta_2 = \pi$ ):

$$F_2 = G_{i2}H_{i2}I_2 + R_2G_{r2}H_{r2}I_2 = G_{i2}H_{i2}I_2(1 - R_2) + R_2H_{i2}I_2 \quad (100)$$

Finally,  $F_3$  is simply equal to:

$$F_3(x, y) = -H_{i0} = -H_{i2}I_2 \quad (101)$$

Summation of these three terms finally yields:

$$F = F_1 + F_2 + F_3 = G_{i1}H_{i1}I_1(1 + R_1) + G_{i2}H_{i2}I_2(1 - R_2) - H_{i2}I_2(1 - R_2) \quad (102)$$

This expression is (again) exactly zero if  $R_1 = -1$  and  $R_2 = 1$  (!).

What we have found now, is that for  $R_1 = 1$  and  $R_2 = -1$  the “rigid-type” boundary condition (22) is satisfied along breakwater 1 and the “cushion-type” boundary condition (49) is satisfied along breakwater 2. This is a consistent result (!). And vice versa, for  $R_1 = -1$  and  $R_2 = 1$ , the “cushion-type” boundary condition (49) is satisfied along breakwater 1 and the “rigid-type” boundary condition (22) is satisfied along breakwater 2.

Hence, for these two situations, we can find a diffraction field that *exactly* satisfies the boundary conditions. For the case of normally incident waves, this has also been proven by Penney and Price (1952). In the present study, we have given a more rigorous proof that is valid for *any* wave direction  $\beta$ . The key reason for this proof to be successful has been the fact that elementary Sommerfeld solutions can be given a phase shift ( $I_1 = \exp\{-i\phi\}$  and  $I_2 = \exp\{i\phi\}$ ), in such a way that these solutions remain consistent with the Helmholtz Equation (21) and the boundary conditions (22) and (49).

Unfortunately, such an exact satisfaction of the boundary conditions cannot be found for cases in which  $R_1 = 1$  and  $R_2 = 1$ , or  $R_1 = -1$  and  $R_2 = -1$ . We will examine now the deviation of the theoretical boundary conditions for those cases. (In the next section, we will present a Matlab script which visualizes the boundary values of  $F$  and  $\partial F / \partial y$  along both breakwaters for any given gap width  $B$ , wave length  $L$  and wave direction angle  $\beta$ ; with the help of this script the magnitude of the deviations from zero can be easily visualized for all possible cases.)

If we consider the case  $R_1 = -1$  and  $R_2 = -1$ , we must revisit the equations (91) and (102). If we substitute  $R_1 = -1$  and  $R_2 = -1$  in those equations, we obtain:

$$F = 2G_{i2}H_{i2}I_2 - 2H_{i2}I_2 \quad (\text{along breakwater 1}) \quad (103)$$

$$F = 2G_{i1}H_{i1}I_1 - 2H_{i1}I_1 \quad (\text{along breakwater 2}) \quad (104)$$

These values of  $F$  are generally non-zero, both at the “lee side” and the “exposed side” of the breakwaters. However, their absolute values quickly approach a zero limit within a wave length  $L$



away from the breakwater head, especially when the angle of wave incidence is in the neighborhood of 90°. For wave incidence angles close to 0° or a multiple of 180°, when the waves are traveling virtually parallel to both breakwaters, larger deviations are found.

If we consider the case  $R_1 = 1$  and  $R_2 = 1$ , we must revisit the equations (86) and (97). If we substitute  $R_1 = 1$  and  $R_2 = 1$  in those equations, we obtain:

$$\frac{\partial F}{\partial y} = 2 \left[ \frac{I_2}{2r_2} \frac{1+i}{2} \exp\{-ikr_2\} \tau_{i2} - G_{i2} H_{i2} I_2 ik \sin \beta \right] + 2H_{i0} ik \sin \beta \quad (\text{breakwater 1}) \quad (105)$$

$$\frac{\partial F}{\partial y} = 2 \left[ -\frac{I_1}{2r_1} \frac{1+i}{2} \exp\{-ikr_1\} \tau_{i1} - G_{i1} H_{i1} I_1 ik \sin \beta \right] + 2H_{i0} ik \sin \beta \quad (\text{breakwater 2}) \quad (106)$$

These values of  $\partial F / \partial y$  are generally non-zero, both at the “lee side” and the “exposed side” of the breakwaters. However, their absolute values quickly approach zero at locations within a wave length  $L$  from the breakwater head, especially when the angle of wave incidence is in the neighborhood of 90°. Even for wave incidence angles close to 0° or a multiple of 180°, when the waves are traveling virtually parallel to both breakwaters, the deviations from zero remain quite limited.

Moreover, in any case these deviations from zero are much smaller than for cases with partial reflection or even without reflection, i.e. when  $0 \leq R_1 < 1$  and  $0 \leq R_2 < 1$ . In practical calculations with breakwaters, very often partial reflection will be the present. This means that for practical purposes it is not essential to exactly satisfy the theoretical boundary conditions for  $R_1 = 1$  and  $R_2 = 1$ , firstly because this idealized case is often not representative for reality, and secondly because the error in the  $R_1 = 1$  and  $R_2 = 1$  case is negligible compared to all other cases, where  $0 \leq R_1 < 1$  and  $0 \leq R_2 < 1$ . Therefore, we conclude that the constructed approximate solution (65) is sufficiently accurate for all practical cases, so we finally have for the breakwater gap case with monochromatic waves:

$$F = (G_{i1} + G_{i2} - 1)H_{i0} + R_1 G_{r1} H_{r1} I_1 + R_2 G_{r2} H_{r2} I_2 \quad (107)$$

## 4.6 Implementation

The formulation outlined in the previous section have been implemented within 2 Matlab scripts, which can be found in Appendices E-F.

The approximate Sommerfeld solution for the breakwater gap case has been implemented in the Matlab script **BreakwaterGap\_Test.m**, which is listed in Appendix E. This script utilizes the Matlab function **Sommerfeld\_PenneyPrice.m** (Appendix B) and the Matlab data file **Cornu.mat**, which have been described before in Section 3.7. The script computes and plots the 2D wave diffraction field around a gap in between two breakwaters (aligned with the x-axis), at constant water depth and for a range of wave directions. The separate solutions for the incident wave and both of the reflected waves are plotted into a figure, as well as the superposition of these waves (which is the final solution). Both of the breakwaters are optional and can be switched off on behalf of an easy comparison with solutions for the semi-infinite breakwater case.

Besides the general script for the breakwater case listed in Appendix E, a slightly more dedicated script has been developed named ***BoundaryConditions\_Gap.m***, which is listed in Appendix F. This script computes and plots the solution of the breakwater gap case along the x-axis only (hence we have 1D output), but it does so for both sides of the two breakwaters considered: the “lee side” and the “exposed side”. The user can specify any combination of wave length, gap width, reflection coefficient of the left breakwater and reflection coefficient of the right breakwater, as well as a wide range of wave direction angles. This script utilizes the Matlab data file ***Cornu.mat*** only.

The script plots the absolute values of the solution  $F$  and its normal derivative  $\partial F / \partial y$  along both breakwaters; these values represent the boundary conditions of the “rigid-type” breakwater and the “cushion-type” breakwater, as has been explained in Section 4.5. The user will soon find out that these boundary conditions are satisfied exactly for the combinations  $R_1 = 1 \wedge R_2 = -1$  and  $R_1 = -1 \wedge R_2 = 1$  (as argued before), but only approximately for the combinations  $R_1 = -1 \wedge R_2 = -1$  or  $R_1 = 1 \wedge R_2 = 1$ , or any other configuration with partial reflection involved. Exploring the entire parameter space should persuade the user that the approximate solution for  $R_1 = 1 \wedge R_2 = 1$  is sufficiently accurate for practical purposes (certainly compared to cases with partial reflection), provided that the gap width is at least one wave length:  $B \geq L$ .

## 5 THIRD STEP: ADDING SPECTRAL AND DIRECTIONAL SPREADING CHARACTERISTICS

### 5.1 Introduction

Having performed the crucial second step, the re-construction of the approximate Sommerfeld solution for the breakwater gap case (for monochromatic waves), for the third step we are able now to follow the instructions given by Goda more closely. Whereas the second step was not described explicitly, Goda et al. (1978) give elaborate instructions about how to incorporate frequency spreading and directional spreading characteristics into the solution for monochromatic waves. Nevertheless, some issues still have to be clarified, namely:

- The correct relation between the significant wave period  $T_{1/3}$  and the “representative” wave length  $L$  within the Goda diffraction diagrams;
- The correct integration boundaries for the directional spreading function and the associated directional wave spectrum;
- The correct dimension and Jacobian of the directional wave spectrum, see Komen (1988).

Following the discussion by Goda et al. (1978), we will consecutively address these issues and find a plausible solution for them.

### 5.2 Frequency spectrum; significant wave height and period

As stated by Goda (and many other authors likewise), “random waves in the sea can be described with a directional wave spectrum under the presumption that random wave profiles are the result of linear superposition of an infinite number of infinitesimal wavelets with various frequencies and directions.”

In order to construct diffraction diagrams for random sea waves, Goda et al. (1978) apply a form of the wave spectrum which has been proposed by Mitsuyasu (1968, 1975). In the present study we will apply this form of the spectrum as well, but it has to be stated that the approach as described by Goda et al. (1978) does not depend on the precise form of the spectrum; any formulation for the frequency and directional spectrum can be used, as long as the physical properties of the spectral formulation (in terms of e.g. significant wave height and peak period) are correct.

We will start out here with the frequency part of the spectrum, which is in fact a variance density spectrum. Mitsuyasu (1968) developed this spectral formulation as a modified Bretschneider’s spectrum. The wave variance density  $S(f)$ , as a function of wave frequency  $f$ , reads:

$$S(f) = 0.257 \cdot H_{1/3}^2 T_{1/3}^{-4} f^{-5} \exp \left\{ -1.03 (T_{1/3} f)^4 \right\}, \quad (108)$$

where  $H_{1/3}$  is the significant wave height and  $T_{1/3}$  is the associated significant wave period, which have to be specified by the user. As can be seen from (108),  $H_{1/3}$  merely determines the *size* (vertical scaling) of the spectrum, whereas  $T_{1/3}$  determines the *shape* (horizontal scaling) of the spectrum. The dimension of  $S(f)$  is  $\text{m}^2/\text{Hz}$ , which is equal to  $\text{m}^2 \cdot \text{s}$ . The total variance content of the spectrum is given by the zero-th moment  $m_0$  of  $S(f)$  (with dimension  $\text{m}^2$ ):

$$m_0 = \int_0^{\infty} S(f) df \quad (109)$$

From this moment the significant wave height  $H_{m_0}$  according to the spectrum can be retrieved:

$$H_{m0} = 4.004 \cdot \sqrt{m_0} \quad (110)$$

Of course, we should have  $H_{m0} = H_{1/3}$  for a properly defined spectrum (!). On this place Goda et al. (1978) just used:

$$H_{m0} = 4\sqrt{m_0}, \quad (111)$$

but Equation (110) is theoretically better and also more accurate here. This can be checked as follows. We choose a range of 100 wave frequencies between 0 and  $8/T_{1/3}$  (with increments of  $0.08/T_{1/3}$ ). If we compute  $H_{m0}$  by numerically integrating  $S(f)$  over  $f$  using the trapezoidal rule, we will find  $H_{m0} \approx 0.9989 \cdot H_{1/3}$  using (111) and  $H_{m0} \approx 0.9999 \cdot H_{1/3}$  using (110), for any value of  $T_{1/3}$ . The corresponding errors are  $\varepsilon = H_{m0} / H_{1/3} - 1 \approx 1 \cdot 10^{-3}$  using (111) and  $\varepsilon \approx 9,8 \cdot 10^{-5}$  using (110). In this study we will apply definition (110) throughout. The peak frequency  $f_p$  and peak period  $T_p$  of the spectrum (108) are equal to:

$$f_p = 1/(1.05 \cdot T_{1/3}) \quad \text{and} \quad T_p = 1.05 \cdot T_{1/3} \quad (112)$$

This is quite a common relation between peak period and significant wave period (see Holthuijsen, 2007). Another expression for a representative wave period is given by:

$$\bar{T}_0 = \sqrt{m_0 / m_2}, \quad \text{with} \quad m_2 = \int_0^{\infty} S(f) f^2 df, \quad (113)$$

where  $m_2$  is the second moment of  $S(f)$  (with dimension  $m^2/s^2$ ). Theoretically,  $\bar{T}_0$  represents the so-called mean zero-crossing period of the wave field, see Rice (1954). This period is *not* equal to the significant wave period, yet in the approach by Goda et al. (1978) expression (113) has an important role. Because  $\bar{T}_0$  is linearly proportional to  $T_{1/3}$  (which is at least true for the Mitsuyasu spectrum given by (108)), we can use  $\bar{T}_0$  as a measure for  $T_{1/3}$ ; this will be applied in Section 5.4.

In order to construct diffraction diagrams for random waves based on (108), the user should specify both  $H_{1/3}$  and  $T_{1/3}$ . However, Goda et al. (1978) do not mention  $T_{1/3}$  in their diagrams; instead, they apply a “representative” wave length  $L$ , which is made dimensionless by the breakwater gap width  $B$ . The relation between  $T_{1/3}$  and  $L$  is not made explicit; it is however likely that these quantities are just directly correlated by means of a dispersion relation, probably for deep-water conditions (because, strikingly, Goda et al. never mention the influence of the water depth at all). Presumably, the authors did assume that applying deep-water wave conditions would lead to conservative results, which may not always be true. (Within near-shore harbours, the influence of limited water depth on incident waves can be expected to be important – and even more if also frequency spreading is present.) Therefore, we propose here to use the general dispersion relation for linear gravity waves, which reads (applied to  $T_{1/3}$  and  $L$ ):

$$\left(\frac{2\pi}{T_{1/3}}\right)^2 = g \left(\frac{2\pi}{L}\right) \tanh\left(\frac{2\pi}{L} d\right), \quad (114)$$

in which  $d$  is the water depth. Furthermore, standard linear diffraction diagrams are usually related to a unit wave height (or unit wave amplitude), so that we can use  $H_{1/3} = 1$  m in general. Hence, we only need to specify  $L$  after all, after which  $T_{1/3}$  follows from (114) and  $H_{1/3}$  is set to unity.

### 5.3 Directional spectrum; correct integration over wave directions

The wave frequency spectrum  $S(f)$  as given in (108) can be extended to a frequency-direction spectrum  $E(f, \beta)$  in the following way:

$$E(f, \beta) = S(f)D(f, \beta), \quad (115)$$

in which  $D(f, \beta)$  is a dimensionless directional spreading function which has been defined by Mitsuyasu (1975). Goda et al. (1978) use the symbol  $\theta$  for the spreading of the wave direction angle, assuming that the dominant wave direction would be  $\theta = 0$ . However, as we want to construct diffraction diagrams for any dominant wave direction angle, we will replace  $\theta$  by  $\beta - \beta_0$ , in which  $\beta$  is the actual direction of a wave component (as defined in Section 3.6) and  $\beta_0$  is the dominant wave direction. The formulation by Mitsuyasu (1975) then reads:

$$D(f, \beta) = D_0 \cos^{2S} \left( \frac{\beta - \beta_0}{2} \right), \quad (116)$$

where:

$$D_0 = \left[ \int_{\beta_0 - \pi}^{\beta_0 + \pi} \cos^{2S} \left( \frac{\beta - \beta_0}{2} \right) d\beta \right]^{-1} \quad (117)$$

and

$$S = \begin{cases} S_{\max} (f / f_p)^5 & \text{for } f \leq f_p \\ S_{\max} (f / f_p)^{-2.5} & \text{for } f \geq f_p \end{cases} \quad (118)$$

The quantity  $D_0$  is a normalization factor, which ensures that the integral of  $D(f, \beta)$  over all possible wave directions is exactly 1 for each separate frequency  $f$ . The function  $S$  determines the width of the spreading function, which is in turn a function of the frequency. Finally, the parameter  $S_{\max}$  determines the over-all width or narrowness of the directional spreading function. Following Goda et al. (1978), we set  $S_{\max} = 10$  for wind waves (short-crested waves, wide spectrum) and  $S_{\max} = 75$  for swell (long-crested waves, narrow spectrum). These standard values for  $S_{\max}$  are in use within the “spreadsheet method” as well.

From (117), it can be seen that integration over the wave directions takes place in between the angles  $\beta_0 - \pi$  and  $\beta_0 + \pi$ . Goda et al. (1978) use the integration boundaries  $\theta_{\min}$  and  $\theta_{\max}$  here, without ever specifying where exactly these boundaries are located. We state here that the integration boundaries  $\beta_0 - \pi$  and  $\beta_0 + \pi$  make the most sense, and they do so in two ways: firstly, integration takes place over *all* possible directions (360°); and secondly, the dominant wave

direction  $\beta_0$  (where  $D(f, \beta)$  has its maximum) is centred exactly half-way the domain of integration.

Similarly to the frequency spectrum, the directional spreading function may be replaced by other forms as well, as long as the requirement  $\int D(f, \beta) d\beta = 1$  is satisfied for each frequency  $f$ .

The energy content of the total frequency-direction spectrum is given by:

$$m_0 = \int_{\beta_0-\pi}^{\beta_0+\pi} \int_0^{\infty} E(f, \beta) df d\beta = \int_{\beta_0-\pi}^{\beta_0+\pi} \int_0^{\infty} S(f) D(f, \beta) df d\beta \quad (119)$$

Because  $\int D(f, \beta) d\beta = 1$  for each frequency, it follows that  $m_0$  according to (119) is exactly equal to  $m_0$  according to Equation (109). Hence, the directional spreading function does not have direct impact on the total amount of incident wave energy.

So far within this section, we have adopted a definition of the frequency-direction spectrum which often occurs in literature (see Holthuijsen, 2007) but which also has a drawback (see Komen, 1988). When making a “polar plot” of  $E(f, \beta)$ , the plotted densities are *not* adequate variance densities, because polar integrals over any spectral (sub)domain are generally *not* equal to Cartesian integrals over the same (sub)domain:  $\iint E(f, \beta) df d\beta \neq \iint E(f, \beta) dx dy$ . The latter is only the case if the polar frequency-direction spectrum is divided by the “Jacobian”  $J = f$ , which follows from a formal transformation between polar coordinates and Cartesian coordinates. This operation defines a modified spectrum  $\hat{E}(f, \beta)$  (with dimension  $m^2 \cdot s^2$  now!):

$$\hat{E}(f, \beta) = \frac{S(f)}{J} D(f, \beta) \quad (120)$$

Now, the total spectral variance content (zero-th moment) of the spectrum is given by:

$$\hat{m}_0 = \int_{\beta_0-\pi}^{\beta_0+\pi} \int_0^{\infty} \hat{E}(f, \beta) f df d\beta = \int_{\beta_0-\pi}^{\beta_0+\pi} \int_0^{\infty} \frac{S(f)}{J} D(f, \beta) f df d\beta \quad (121)$$

Now, instead of  $df d\beta$ , the spectrum is thus integrated over  $f df d\beta$ , which is the formally correct way to integrate over polar coordinates. Because  $J = f$ , we can readily see from (121) that  $\hat{m}_0 = m_0$ . Furthermore, we have accomplished that  $\iint \hat{E}(f, \beta) f df d\beta = \iint \hat{E}(f, \beta) dx dy$  for each spectral (sub)domain. In this way, the physical interpretation of integrals over “polar plots” has become much more straightforward.

Why, then, should one maintain the original definition (115) and (119), instead of the new definition (120)-(121)? The advantage of (120)-(121) is the simpler geometric interpretation of polar plots and their integrals. However, polar coordinates involving a frequency  $f$  do not make sense (!). Physically, a sensible polar coordinate in radial direction is given by either a distance  $r$  or a wave number  $k$ , and *not* by a frequency (which contains the dimension of time). Of course, when defining a spectrum in terms of wave numbers it is fully appropriate to adopt definition (120)-(121), with the Jacobian equal to the polar wave number ( $J = k = \sqrt{k_x^2 + k_y^2}$ ) and polar integrals taken over the integrand product  $k dk d\beta$ . In terms of frequency spectra, however, it is physically more correct to

maintain the original definition (115) and (119). The latter will be done here, in agreement with Goda et al. (1978) and Holthuijsen (2007).

In practice, both definitions are acceptable and workable; both definitions appear in literature, as mentioned by Komen (1988) in the WAM Newsletter of July 1988. When comparing polar plots, one should make sure that both are defined in the same way. When in doubt, one should always check the dimensions.

## 5.4 Spectral calculation of wave diffraction

Throughout Chapters 3 and 4, we have considered a single monochromatic incident wave with a certain frequency  $f$  (or a wave length  $L$  accordingly) and a wave direction  $\beta$ . For each  $f$  and  $\beta$ , the Sommerfeld solution is given by a complex wave function  $F(x, y)$ ; in order to express the dependence of  $F(x, y)$  on the parameter values  $f$  and  $\beta$ , we will write  $F(f, \beta|x, y)$  from here onward.

The approach in which waves are described using a variance density spectrum or energy density spectrum does not use any phase information; it is a “phase-averaged” approach. Therefore, in order to perform a spectral calculation of wave diffraction, we will discard the phase information within  $F(f, \beta|x, y)$  and proceed with the amplitude information only. We define the *diffraction coefficient*  $K_d(f, \beta|x, y) = |F(f, \beta|x, y)|$ , which is the amplitude of the diffracted wave  $F(x, y)$  for a given geometry and given values of  $f$  and  $\beta$ , where the amplitude of the *incident wave* is equal to 1 (!). Because wave diffraction is considered here as a linear problem, *any incident wave amplitude* (or wave height) may be multiplied by  $K_d(f, \beta|x, y)$  in order to find the diffracted wave amplitude (or wave height) on each location  $(x, y)$  for a given geometry.

The correct way of incorporating spectral information into wave diffraction problems is to multiply the frequency-direction spectrum of the *incident wave*  $E_i(f, \beta)$  by the diffraction coefficient  $K_d(f, \beta|x, y)$ , for a grid of  $(f, \beta)$ -values of sufficient resolution, and integrate the result. In this way, we find the spectral energy content of the diffracted wave for each location  $(x, y)$ :

$$m_{0;d}(x, y) = \int_{\beta_0-\pi}^{\beta_0+\pi} \int_0^\infty E_i(f, \beta) K_d^2(f, \beta|x, y) df d\beta \quad (122)$$

By integration over the wave directions only and not over the frequency domain, we find the frequency spectrum of the diffracted wave:

$$S_d(f|x, y) = \int_{\beta_0-\pi}^{\beta_0+\pi} E_i(f, \beta) K_d^2(f, \beta|x, y) d\beta \quad (123)$$

And by not integrating at all, of course, we find a quantity which represents the full frequency-direction spectrum of the diffracted wave:  $E_d(f, \beta|x, y) = E_i(f, \beta) K_d^2(f, \beta|x, y)$ . This quantity however does not make much physical sense, because the direction  $\beta$  is actually the direction of the *incident wave*, whereas the local direction of the diffracted wave can be *any* direction, so to speak. Goda et al. (1978) put it like this: “The spectrum of diffracted waves is given here in the form of a frequency spectrum only, because the directional spreading of diffracted waves

is limited by the aperture of the breakwater gap looked from the point  $(x, y)$ ". That is not all, however: it is just as important (and probably much more important) that the local wave direction of the diffracted wave can be totally different from that of the incident wave, because that is the very nature of diffraction (!).

Therefore, we will restrict ourselves here to the frequency spectrum of the diffracted wave, given by (123), and the energy content of the diffracted wave given by (122). Of course, these are related in the following way:

$$m_{0;d}(x, y) = \int_0^{\infty} S_d(f|x, y) df \quad (124)$$

The corresponding significant wave height  $H_{m0;d}$  of the diffracted wave reads:

$$H_{m0;d}(x, y) = 4.004 \cdot \sqrt{m_{0;d}(x, y)} \quad (125)$$

The energy content of the incident wave is given by:

$$m_{0;i} = \int_{\beta_0 - \pi}^{\beta_0 + \pi} \int_0^{\infty} E_i(f, \beta) df d\beta \quad (126)$$

And the corresponding significant wave height  $H_{m0;i}$  of the incident wave reads:

$$H_{m0;i} = 4.004 \cdot \sqrt{m_{0;i}} = H_{1/3} \quad (127)$$

From these quantities we can derive an *effective diffraction coefficient*  $K_{d;eff}(x, y)$ , a 2D field of real numbers which represent the ratio between incident significant wave height and local diffracted significant wave height:

$$K_{d;eff}(x, y) = \frac{H_{m0;d}(x, y)}{H_{m0;i}} = \sqrt{\frac{m_{0;d}(x, y)}{m_{0;i}}} \quad (128)$$

The values of  $K_{d;eff}$  as a function of the location  $(x, y)$  are the most important output data within standard diffraction diagrams.

Besides an effective diffraction coefficient (i.e. incident/diffracted wave *height* ratio), we can also derive an effective incident/diffracted wave *period* ratio. To this end, we compute the second moment of the incident wave spectrum:

$$m_{2;i} = \int_{\beta_0 - \pi}^{\beta_0 + \pi} \int_0^{\infty} E(f, \beta) f^2 df d\beta \quad (129)$$

and of the diffracted wave spectrum:

$$m_{2;d}(x, y) = \int_0^{\infty} S_d(f|x, y) f^2 df \quad (130)$$

Now we can compute the mean zero-crossing period of both wave field spectra, see Rice (1954):

$$\bar{T}_{0;i} = \sqrt{m_{0;i} / m_{2;i}} \quad (131)$$



$$\bar{T}_{0;d}(x, y) = \sqrt{m_{0;d}(x, y) / m_{2;d}(x, y)} \quad (132)$$

It has already been stated that  $\bar{T}_0$  is proportional to  $T_{1/3}$  for the present Mitsuyasu spectrum, and probably for almost all properly defined wave frequency spectra. If we now assume that this property is also valid for the diffracted waves, the ratio  $\bar{T}_{0;d}(x, y) / \bar{T}_{0;i}$  is virtually equal to the ratio  $T_{1/3;d}(x, y) / T_{1/3;i}$  of the significant wave periods of the incident and diffracted wave. In this way we can define an *effective wave period ratio*  $K_{d;T}(x, y)$ :

$$K_{d;T}(x, y) = \frac{T_{1/3;d}(x, y)}{T_{1/3;i}} = \sqrt{\frac{m_{0;d}(x, y) / m_{2;d}(x, y)}{m_{0;i} / m_{2;i}}} \quad (133)$$

The values of  $K_{d;T}$  as a function of the location  $(x, y)$  form another important output quantity within standard diffraction diagrams.

## 5.5 Implementation

The formulations outlined in the previous sections have been implemented within 2 Matlab scripts (for the semi-infinite breakwater case and for the breakwater gap case, respectively), which can be found in Appendices G-J.

The Goda-type diffraction diagrams for the case of a semi-infinite breakwater and a breakwater gap can be constructed using the Matlab scripts **GodaDiagram\_SemilInfiniteBreakwater.m** and **GodaDiagram\_BreakwaterGap.m**, which can be found in the Appendices I and J respectively.

These scripts once more utilize the Matlab function **Sommerfeld\_PenneyPrice.m** (Appendix B) and the Matlab data file **Cornu.mat**, which have been described before in Section 3.7. Furthermore, both scripts use the small Matlab function **WaveNumber.m**, which computes a wave number based on a water depth and a wave period using Newton-Raphson iteration. This function can be found in Appendix H.

The scripts in Appendices I and J compute and plot the 2D wave diffraction field around a semi-infinite breakwater (at any location and orientation) or a gap in between two breakwaters (aligned with the x-axis), including directional spreading and frequency spreading, at constant water depth and for only one dominant wave direction at a time. The latter forms a contrast with the previous scripts in Appendices A and E, in which a large range of (monochromatic) wave conditions could be simulated at once. We have chosen not to add this option to the scripts in Appendices I and J, because the computational cost of adding directional spreading and frequency spreading to these computations is quite large already. Sufficient resolution, both in terms of frequency and wave directions, is required in order to build proper spectra for diffracted waves throughout the computational domain.

The results of both scripts are plotted into a figure and can be saved to a Matlab file by the user. The two graphs plotted in the output figure represent the effective *amplitude* ratio of the incident and diffracted waves as a function of location (left panel), resp. the effective *period* ratio of the incident and diffracted waves as a function of location (right panel).

Besides the general scripts listed in Appendices I and J, a slightly more dedicated script has been developed named **SpectralIntegral.m**, which is listed in Appendix H. This script compares many ways in which the standard 1D and 2D frequency-direction spectra proposed by Goda et al. (1978) and Mitsuyasu (1968, 1975) can be integrated numerically. In this way it is checked whether the resulting

spectrum has the correct wave energy content, and which role is exactly played by the polar coordinate Jacobian  $J = f$  (see Komen, 1988).

## 6 SUMMARY AND CONCLUSIONS

Throughout Chapters 3, 4 and 5 of this report, the three steps described in Chapter 2 were performed to re-construct the diffraction diagrams as described by Goda et al. (1978).

- 1) The analytical Sommerfeld (1896) solutions as described by Penney and Price (1952) and Goda et al. (1978) for the semi-infinite breakwater case have been compared and unified, and they have been adapted to make them robust for any wave direction and any orientation of the breakwater.
- 2) Based on the exact solution emerging from step 1), an approximate solution has been constructed which holds for the breakwater gap case with full reflection (reflection coefficients of both breakwaters equal to 1). Because this step was not described by Goda et al. (1978), it had to be re-constructed.
- 3) The final step has been to incorporate frequency spreading and directional spreading to the monochromatic wave solutions of step 1) and 2), in order to find spectral properties of wave diffraction patterns as well as effective significant wave heights and wave periods ratios of the diffracted and incident wave.

All three steps have been sufficiently documented by the Matlab scripts which are listed in the Appendices A-J. The scripts in Appendices I and J can be used to reproduce the Goda diagrams “with one press on the green button”, as desired in Chapter 1. Moreover, we have the possibility now to produce Goda diagrams:

- 1) At much larger accuracy (much more than only 1 decimal place);
- 2) At any desirable spatial resolution;
- 3) For any breakwater orientation and reflection coefficient (though only approximately for the breakwater gap case);
- 4) For any dominant wave direction, whereas the original Goda diagrams only accounted for normal wave incidence.

Furthermore, we assume to have solved the mystery which role was exactly played by the water depth to construct the original Goda diagrams (probably none).

The general approach behind the construction of diffraction diagrams like those of Goda et al. (1978) does however not depend on the precise mathematical tools used in this study: neither on the exact analytical Sommerfeld (1896) solution for one specific geometry, not on the exact form of the frequency-direction spectrum as proposed by Mitsuyasu (1968, 1975).

Possible extensions of the current approach are:

- 1) Firstly, instead of analytical solutions, one is fully free to apply solutions of numerical models, which in turn allows the user the flexibility to generate solutions for an arbitrary (non-schematized) geometry.
- 2) Secondly, one may choose any arbitrary but physically relevant wave spectrum instead of the Mitsuyasu spectrum applied here.

In view of this more generic approach, the central equation within the approach by Goda et al. (1978) is actually Equation (123), which described how spectral information can be incorporated into initially monochromatic wave diffraction solutions. Whether or not the user chooses to apply totally different numerical solutions and/or totally different wave spectra, this equation (which is actually the very first Equation within the original Goda paper) remains valid at all times.

In this study, the original diffraction tables from the “spreadsheet method” (RIKZ, 2004) have been reproduced using the developed Matlab scripts. With respect to these original tables, the only difference with the new tables is the enhanced accuracy (2 decimal places instead of 1, and more decimal places can always be computed if desirable) and probably a larger spectral resolution as well. Furthermore, it is noticeable that the original tables from the “spreadsheet method” have been

constructed graphically by reading and interpolating values from the original diagrams by Goda et al. (1978), whereas the new diffraction table values have been computed directly from the Matlab script output. The new tables are found within the Excel sheet "Diffracteddiagrammen\_GODA\_09072014.xlsx".

## REFERENCES

- Boshek, M.R. (2009). Reflection and Diffraction around Breakwaters. *M.Sc. Thesis*, Delft University of Technology, Delft, June 2009.
- Daemrich, K.-F., Kohlhase, S. (1978). Influence of Breakwater-Reflection on Diffraction. *Proceedings of the 16th International Conference on Coastal Engineering (Hamburg)*, New York, ASCE, pp. 651-663.
- Goda, Y., Takayama, T., Suzuki, Y. (1978). Diffraction diagrams for directional random waves. *Proceedings of the 16th International Conference on Coastal Engineering (Hamburg)*, New York, ASCE, pp. 629-650.
- Holthuijsen, L.H. (2007). *Waves in Oceanic and Coastal Waters*. Cambridge University Press.
- Komen, G.J. (1988). What is the frequency-directional spectrum? *WAM Group Newsletter*, No. 6, July 1988.
- Mitsayasu, H. (1968). On the growth of the spectrum of wind-generated waves (I), *Reports of Research Institute for Applied Mechanics*, Kyushu University, Vol. 16, No. 55, pp. 459-482.
- Mitsayasu, H., Tasai, F., Suhara, T., Mizuno, S., Ohkusu, M., Honda, T., Rikiishi, K. (1975). Observations of the directional spectrum of ocean waves using a cloverleaf buoy. *Journal of Physical Oceanography*, Vol. 5, No. 10, pp. 750-760.
- Penney, W.G., Price, A.T. (1952). The diffraction theory of sea waves and the shelter afforded by breakwaters. *Philosophical Transactions of the Royal Society of London*, A. 244, pp.236-253.
- Rice, S.O. (1954). Mathematical analysis of random noise. *Selected Papers on Noise and Stochastic Processes*, ed. N. Wax, New York, Dover Publications Inc., pp. 133-294.
- Rijkstinstituut voor Kust en Zee/ RIKZ (2004). Golfbelastingen in havens en afgeschermede gebieden. Een gedetailleerde methode voor het bepalen van golfbelastingen voor het toetsen van waterkeringen, February 2004.
- Sommerfeld, A. (1896). Mathematische Theorie der Diffraction. *Math. Ann.* 47, p. 317.

## APPENDICES

### Appendix A: Matlab script SemilInfiniteBreakwater\_Test.m

```

% Script SemiInfiniteBreakwater_Test.m
%
% This script computes the 2D wave diffraction around a semi-infinite
% breakwater, at constant water depth and for a range of wave directions,
% using the analytical Sommerfeld (1896) solution as given by Penney and
% Price (1952) and by Goda et al. (1978). The two resulting solutions are
% automatically compared for each wave direction and the Penney and Price
% (1952) results are subsequently plotted into a figure.
%
% (c) H. Talstra, Svasek Hydraulics, Rotterdam, The Netherlands, 14 February 2014
%
% The script makes use of the following Matlab .mat-file:
%   Cornu.mat
% and the following two Matlab functions:
%   Sommerfeld_PenneyPrice.m
%   Sommerfeld_Goda.m
%
% N.B. Within the context of this Matlab script, no information about
% wave height/amplitude, wave period and water depth is required. The
% resulting wave function F is valid for an incident wave with unit amplitude
% and the user may multiply it by the actual incident wave height/amplitude. The
% necessary information about wave period and water depth is already accounted
% for by the specified wave length/wave number (via the dispersion relation).

%% Cleanup of workspace
clear all;
close all;
clc;

%% Input parameters
X=(-1200:10:1200)'; % Range of x-coordinates for global grid [m]
Y=(-1000:10:1000)'; % Range of y-coordinates for global grid [m]
[x,y]=meshgrid(X,Y); % Construct global Cartesian grid (x,y) [m]
x0=300; % Global x-coordinates of the breakwater head [m]
y0=400; % Global y-coordinates of the breakwater head [m]
breakwaterdir=37; % Orientation of the breakwater [degrees]
wavedir_range=(-720:15:720); % Global wave direction range [degrees]
L=100; % Wave length [m]
load Cornu.mat; % Import Cornu Spiral data

%% Convert degrees to radians
delta=breakwaterdir*pi/180;
beta_range=wavedir_range*pi/180;

%% Loop over all imposed wave directions
uit=figure('Position',[200 50 1500 900]); % Initialize figure
d=sqrt((max(X)-min(X)).^2+(max(Y)-min(Y)).^2); % Initialize figure
n=length(beta_range); % Initialize maximum difference registration
maxdiff_F = zeros(n,1); % Initialize maximum difference registration
maxdiff_Fi = zeros(n,1); % Initialize maximum difference registration
maxdiff_Fr = zeros(n,1); % Initialize maximum difference registration
maxdiff_Gi = zeros(n,1); % Initialize maximum difference registration
maxdiff_Gr = zeros(n,1); % Initialize maximum difference registration
maxdiff_Hi = zeros(n,1); % Initialize maximum difference registration
maxdiff_Hr = zeros(n,1); % Initialize maximum difference registration
maxdiff_sigma1 = zeros(n,1); % Initialize maximum difference registration
maxdiff_sigma2 = zeros(n,1); % Initialize maximum difference registration
j=0; % Initialize counter
for beta=beta_range
    j=j+1; % Update counter
    % Compute and store solution according to Penney and Price (1952)
    [F,Fi,Fr,Gi,Gr,Hi,Hr,sigma1,sigma2]=Sommerfeld_PenneyPrice(x,y,x0,y0,delta,beta,L,Cornu);
    PP.F = F; % Total wave field
    PP.Fi = Fi; % Incident wave field
    PP.Fr = Fr; % Reflected wave field
    PP.Gi = Gi; % Fresnel integral of the incident wave field
    PP.Gr = Gr; % Fresnel integral of the reflected wave field
    PP.Hi = Hi; % Undisturbed incident wave field
    PP.Hr = Hr; % Undisturbed reflected wave field
    PP.sigma1 = sigma1; % Integration boundaries for the incident wave field
    PP.sigma2 = sigma2; % Integration boundaries for the reflected wave field
    clear F Fi Fr Gi Gr Hi Hr sigma1 sigma2;
    % Compute and store solution according to Goda (1978)

```

```

[F,Fi,Fr,Gi,Gr,Hi,Hr,gamma1,gamma2]=Sommerfeld_Goda(x,y,x0,y0,delta,beta,L,Cornu);
Goda.F = F; % Total wave field
Goda.Fi = Fi; % Incident wave field
Goda.Fr = Fr; % Reflected wave field
Goda.Gi = Gi; % Fresnel integral of the incident wave field
Goda.Gr = Gr; % Fresnel integral of the reflected wave field
Goda.Hi = Hi; % Undisturbed incident wave field
Goda.Hr = Hr; % Undisturbed reflected wave field
Goda.gamma1 = gamma1; % Integration boundaries for the incident wave field
Goda.gamma2 = gamma2; % Integration boundaries for the reflected wave field
clear F Fi Fr Gi Gr Hi Hr gamma1 gamma2;
% Compute largest differences between both solutions
maxdiff_F(j) = max(max(abs(PP.F-Goda.F)));
maxdiff_Fi(j) = max(max(abs(PP.Fi-Goda.Fi)));
maxdiff_Fr(j) = max(max(abs(PP.Fr-Goda.Fr)));
maxdiff_Gi(j) = max(max(abs(PP.Gi-Goda.Gi)));
maxdiff_Gr(j) = max(max(abs(PP.Gr-Goda.Gr)));
maxdiff_Hi(j) = max(max(abs(PP.Hi-Goda.Hi)));
maxdiff_Hr(j) = max(max(abs(PP.Hr-Goda.Hr)));
maxdiff_sigma1(j) = max(max(abs(PP.sigma1-Goda.gamma1)));
maxdiff_sigma2(j) = max(max(abs(PP.sigma2-Goda.gamma2)));
% Write largest differences to screen
disp(['Wave direction = ',num2str(wavedir_range(j)),'\n']);
disp('Largest difference between both solutions:');
disp(num2str([...
    maxdiff_F(j),maxdiff_Fi(j),maxdiff_Fr(j),maxdiff_Gi(j),maxdiff_Gr(j),...
    maxdiff_Hi(j),maxdiff_Hr(j),maxdiff_sigma1(j),maxdiff_sigma2(j)]));
disp(' ');
% Plot results: update figure
subplot(3,4,1);
pcolor(x,y,abs(PP.Fi));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'-g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\n'],'fontsize',12,'fontweight','b');
hold off;
xlabel('Amplitude','fontsize',12,'fontweight','b');
ylabel('Incident wave','fontsize',12,'fontweight','b');
subplot(3,4,2);
pcolor(x,y,angle(PP.Fi));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'-g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\n'],'fontsize',12,'fontweight','b');
hold off;
xlabel('Phase','fontsize',12,'fontweight','b');
ylabel('Incident wave','fontsize',12,'fontweight','b');
subplot(3,4,3);
pcolor(x,y,PP.sigma1);
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'-g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\n'],'fontsize',12,'fontweight','b');
hold off;
xlabel('\sigma_1','fontsize',12,'fontweight','b');
ylabel('Incident wave','fontsize',12,'fontweight','b');
subplot(3,4,4);
pcolor(x,y,sign(PP.sigma1));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'-g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\n'],'fontsize',12,'fontweight','b');
hold off;
xlabel('sign(\sigma_1)','fontsize',12,'fontweight','b');
ylabel('Incident wave','fontsize',12,'fontweight','b');
subplot(3,4,5);
pcolor(x,y,abs(PP.Fr));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'-g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\n'],'fontsize',12,'fontweight','b');
hold off;
xlabel('Amplitude','fontsize',12,'fontweight','b');
ylabel('Reflected wave','fontsize',12,'fontweight','b');

```

```

subplot(3,4,6);
pcolor(x,y,angle(PP.Fr));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'--g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\circ'],'fontsize',12,'fontweight','b');
hold off;
xlabel('Phase','fontsize',12,'fontweight','b');
ylabel('Reflected wave','fontsize',12,'fontweight','b');
subplot(3,4,7);
pcolor(x,y,PP.sigma2);
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'--g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\circ'],'fontsize',12,'fontweight','b');
hold off;
xlabel('\sigma_2','fontsize',12,'fontweight','b');
ylabel('Reflected wave','fontsize',12,'fontweight','b');
subplot(3,4,8);
pcolor(x,y,sign(PP.sigma2));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'--g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\circ'],'fontsize',12,'fontweight','b');
hold off;
xlabel('sign(\sigma_2)','fontsize',12,'fontweight','b');
ylabel('Reflected wave','fontsize',12,'fontweight','b');
subplot(3,4,9);
pcolor(x,y,abs(PP.F));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'--g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\circ'],'fontsize',12,'fontweight','b');
hold off;
xlabel('Amplitude','fontsize',12,'fontweight','b');
ylabel('Total wave','fontsize',12,'fontweight','b');
subplot(3,4,10);
pcolor(x,y,angle(PP.F));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'--g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\circ'],'fontsize',12,'fontweight','b');
hold off;
xlabel('Phase','fontsize',12,'fontweight','b');
ylabel('Total wave','fontsize',12,'fontweight','b');
subplot(3,4,11);
pcolor(x,y,PP.sigmal-Goda.gammal);
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'--g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\circ'],'fontsize',12,'fontweight','b');
hold off;
xlabel('Difference \sigma_1-\gamma_1','fontsize',12,'fontweight','b');
ylabel('Total wave','fontsize',12,'fontweight','b');
subplot(3,4,12);
pcolor(x,y,PP.sigma2-Goda.gamma2);
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta)],[y0 y0+d*sin(beta)'],'--g','linewidth',2);hold on;
plot([x0 x0+d*cos(delta)],[y0 y0+d*sin(delta)'],'-k','linewidth',2);hold on;
title(['\beta = ',num2str(wavedir_range(j)),'\circ'],'fontsize',12,'fontweight','b');
hold off;
xlabel('Difference \sigma_2-\gamma_2','fontsize',12,'fontweight','b');
ylabel('Total wave','fontsize',12,'fontweight','b');
drawnow;
% End of plot
end;
% End of loop
%% End of Matlab script

```



## Appendix B: Matlab function Sommerfeld\_PenneyPrice.m

```

function
[F,Fi,Fr,Gi,Gr,Hi,Hr,sigma1,sigma2]=Sommerfeld_PenneyPrice(x,y,x0,y0,delta,beta,L,Cornu)
% function
[F,Fi,Fr,Gi,Gr,Hi,Hr,sigma1,sigma2]=Sommerfeld_PenneyPrice(x,y,x0,y0,delta,beta,L,Cornu)
%
% This function computes the analytical Sommerfeld (1896) solution a of 2D wave
% diffraction field around a semi-infinite breakwater with arbitrary location and
% orientation, at constant water depth.
% The Sommerfeld formulation by Penney and Price (1952) is used, with a
% correction by Talstra (2014) on behalf of validity for all possible wave
% directions.
%
% (c) H. Talstra, Svasek Hydraulics, Rotterdam, The Netherlands, 14 February 2014
%
% The following input parameters are needed:
%   x,y    [m]   Global x- and y-coordinates of the computational grid
%           (may be either a structured or unstructured grid)
%   x0,y0  [m]   Global x- and y-coordinates of the breakwater head
%   delta  [rad] Orientation of the breakwater
%           (as seen from the breakwater head toward infinity)
%   beta   [rad] Global wave direction
%           (direction that the wave train is heading toward, i.e.
%           the orientation of the "shadow line" as seen from the
%           breakwater head)
%   L      [m]   Wave length
%
% The following output is generated:
%   F      [m]   Total wave field (complex function)
%   Fi     [m]   Incident wave field (complex function)
%   Fr     [m]   Reflected wave field (complex function)
%   Gi     [-]   Fresnel integral of the incident wave field (complex function)
%   Gr     [-]   Fresnel integral of the reflected wave field (complex function)
%   Hi     [-]   Undisturbed incident wave field (complex function)
%   Hr     [-]   Undisturbed reflected wave field (complex function)
%   sigma1 [-]   Integration boundaries for the incident wave field
%   sigma2 [-]   Integration boundaries for the reflected wave field
%
% N.B. Within the context of this Matlab function, no information about
% wave height/amplitude, wave period and water depth is required. The
% resulting wave function F is valid for an incident wave with unit amplitude
% and the user may multiply it by the actual incident wave height/amplitude. The
% necessary information about wave period and water depth is already accounted
% for by the specified wave length/wave number (via the dispersion relation).

%% Initialization
k=2*pi/L; % Compute wave number [1/m]
theta0=beta-delta; % Compute local wave direction [rad]
xx=(x-x0).*cos(delta)+(y-y0).*sin(delta); % Local Cartesian coordinate system (xx,yy)
yy=-(x-x0).*sin(delta)+(y-y0).*cos(delta); % Local Cartesian coordinate system (xx,yy)
r=sqrt(xx.*xx+yy.*yy); % Local polar coordinate system (r,theta)
theta=atan2(yy,xx); % Local polar coordinate system (r,theta)
jj=(theta<0); theta(jj)=theta(jj)+2*pi; clear jj; % Make sure that theta is always in
between 0 and 2*pi (!)

%% Compute integration boundaries
sigma1= 2*sqrt(k*r/pi).*sin(0.5*(theta-theta0)).*sign(cos(0.5*(theta0))); % Incident wave
field
sigma2=-2*sqrt(k*r/pi).*sin(0.5*(theta+theta0)).*sign(cos(0.5*(theta0))); % Reflected wave
field

%% Compute Fresnel integrals (with the help of the Cornu Spiral)
Gi=0.5+interp1(Cornu.sigma,Cornu.G,sigma1); % Incident wave field
Gr=0.5+interp1(Cornu.sigma,Cornu.G,sigma2); % Reflected wave field

%% Compute complex wave functions of the undisturbed wave
Hi=exp(-i*k*r.*cos(theta-theta0)); % Incident wave field
Hr=exp(-i*k*r.*cos(theta+theta0)); % Reflected wave field

%% Compute final wave functions
Fi=Gi.*Hi; % Incident wave field
Fr=Gr.*Hr; % Reflected wave field
F=Fi+Fr; % Total wave field

%% End of Matlab function

```

## Appendix C: Matlab function Sommerfeld\_Goda.m

```

function [F,Fi,Fr,Gi,Gr,Hi,Hr,gamma1,gamma2]=Sommerfeld_Goda(x,y,x0,y0,delta,beta,L,Cornu)
% function
[F,Fi,Fr,Gi,Gr,Hi,Hr,gamma1,gamma2]=Sommerfeld_Goda(x,y,x0,y0,delta,beta,L,Cornu)
%
% This function computes the analytical Sommerfeld (1896) solution a of 2D wave
% diffraction field around a semi-infinite breakwater with arbitrary location and
% orientation, at constant water depth.
% The Sommerfeld formulation by Goda et al. (1978) is used, with a
% correction by Talstra (2014) on behalf of validity for all possible wave
% directions.
%
% (c) H. Talstra, Svasek Hydraulics, Rotterdam, The Netherlands, 14 February 2014
%
% The following input parameters are needed:
%   x,y    [m]   Global x- and y-coordinates of the computational grid
%             (may be either a structured or unstructured grid)
%   x0,y0  [m]   Global x- and y-coordinates of the breakwater head
%   delta  [rad] Orientation of the breakwater
%             (as seen from the breakwater head toward infinity)
%   beta   [rad] Global wave direction
%             (direction that the wave train is heading toward, i.e.
%             the orientation of the "shadow line" as seen from the
%             breakwater head)
%   L      [m]   Wave length
%
% The following output is generated:
%   F      [m]   Total wave field (complex function)
%   Fi     [m]   Incident wave field (complex function)
%   Fr     [m]   Reflected wave field (complex function)
%   Gi     [-]   Fresnel integral of the incident wave field (complex function)
%   Gr     [-]   Fresnel integral of the reflected wave field (complex function)
%   Hi     [-]   Undisturbed incident wave field (complex function)
%   Hr     [-]   Undisturbed reflected wave field (complex function)
%   gamma1 [-]   Integration boundaries for the incident wave field
%   gamma2 [-]   Integration boundaries for the reflected wave field
%
% N.B. Within the context of this Matlab function, no information about
% wave height/amplitude, wave period and water depth is required. The
% resulting wave function F is valid for an incident wave with unit amplitude
% and the user may multiply it by the actual incident wave height/amplitude. The
% necessary information about wave period and water depth is already accounted
% for by the specified wave length/wave number (via the dispersion relation).

%% Initialization
k=2*pi/L; % Compute wave number [1/m]
theta=beta-delta+pi; % Compute local wave direction [rad]
xx=(x-x0).*cos(delta)+(y-y0).*sin(delta); % Local Cartesian coordinate system (xx,yy)
yy=(x-x0).*sin(delta)+(y-y0).*cos(delta); % Local Cartesian coordinate system (xx,yy)
r=sqrt(xx.*xx+yy.*yy); % Local polar coordinate system (r,alpha)
alpha=atan2(yy,xx); % Local polar coordinate system (r,alpha)
jj=(alpha<0); alpha(jj)=alpha(jj)+2*pi; clear jj; % Make sure that alpha is always in
between 0 and 2*pi (!)

%% Compute integration boundaries
gamma1=sqrt(4*k*r/pi).*cos(0.5*(alpha-theta)).*sign(sin(0.5*(theta))); % Incident wave
field
gamma2=sqrt(4*k*r/pi).*cos(0.5*(alpha+theta)).*sign(sin(0.5*(theta))); % Reflected wave
field

%% Compute Fresnel integrals (with the help of the Cornu Spiral)
Gi=0.5+interpl(Cornu.sigma,Cornu.G,gamma1); % Incident wave field
Gr=0.5+interpl(Cornu.sigma,Cornu.G,gamma2); % Reflected wave field

%% Compute complex wave functions of the undisturbed wave
Hi=exp(i*k*r.*cos(alpha-theta)); % Incident wave field
Hr=exp(i*k*r.*cos(alpha+theta)); % Reflected wave field

%% Compute final wave functions
Fi=Gi.*Hi; % Incident wave field
Fr=Gr.*Hr; % Reflected wave field
F=Fi+Fr; % Total wave field

% End of Matlab function

```

## Appendix D: Matlab script CornuSpiral.m

```

% Script CornuSpiral.m
%
% This script computes and plots the double-sided Cornu Spiral which is needed for
% the construction of the analytical Sommerfeld solution of a 2D wave diffraction
% field around breakwaters.
%
% (c) H. Talstra, Svasek Hydraulics, Rotterdam, The Netherlands, 14 February 2014

%% Cleanup of workspace
clear all;
close all;
clc;

%% Construct Fresnel integral
W=(-10000000000 -100:0.01:100 10000000000)'; % Define phase difference vector
sigma=sign(W).*sqrt(4*abs(W)); % Compute integration boundary vector (with correct signs)
F=W*0; % Initialize Fresnel integral
n=length(W);
for k=1:n % Fill Fresnel integral
    t=sigma(k); % with the help of numerical integration
    u=(0:0.001:1)*t; % (trapezoidal rule)
    F(k)=trapez(u,exp(0.5*pi*i*u.*u));
end;
G=0.5*(1+i)*conj(F); % Construct final complex function for Cornu Spiral
F(1)=-0.5-i*0.5; % Set limit value for Fresnel integral (at minus infinity)
F(end)=0.5+i*0.5; % Set limit value for Fresnel integral (at plus infinity)
G(1)=-0.5; % Set limit value for Cornu Spiral (at minus infinity)
G(end)=0.5; % Set limit value for Cornu Spiral (at plus infinity)

%% Plot results to figure
figure;
subplot(2,2,1);
plot(conj(F),'-b');hold;plot([-0.5 0.5],[-0.5 0.5],'or');hold;
subplot(2,2,2);
plot(real(F),-imag(F),'-g');hold;plot([-0.5 0.5],[-0.5 0.5],'or');hold;
subplot(2,2,3:4);
plot(G,'-r');hold;plot([-0.5 0.5],[0 0],'or');hold;
disp([F([1 end]) abs(F([1 end]))]);
disp([G([1 end]) abs(G([1 end]))]);

%% Save results to Matlab .mat-file
Cornu.sigma=sigma;
Cornu.G=G;
save Cornu.mat Cornu;

%% End of Matlab script

```

## Appendix E: Matlab script BreakwaterGap\_Test.m

```

% Script BreakwaterGap_Test.m
%
% This script computes and plots the 2D wave diffraction around a gap in
% between two breakwaters, at constant water depth and for a range of wave
% directions, using the analytical Sommerfeld (1896) solution for 2D
% diffraction around a semi-infinite breakwater. This exact solution is
% extended and modified to construct an approximate solution for the case
% with a breakwater gap. The separate solutions for the incident wave and
% both of the reflected waves are plotted into a figure, as well as the
% superposition of these waves (which is the final solution). Both of the
% breakwaters are optional and can be switched off on behalf of an easy
% comparison with solutions for the semi-infinite breakwater case.
%
% (c) H. Talstra, Svasek Hydraulics, Rotterdam, The Netherlands, 26 February 2014
%
% The script makes use of the following Matlab .mat-file:
%   Cornu.mat
% and the following Matlab function:
%   Sommerfeld_PenneyPrice.m
%
% N.B. Within the context of this Matlab script, no information about
% wave height/amplitude, wave period and water depth is required. The
% resulting wave function F is valid for an incident wave with unit amplitude
% and the user may multiply it by the actual incident wave height/amplitude. The
% necessary information about wave period and water depth is already accounted
% for by the specified wave length/wave number (via the dispersion relation).

%% Cleanup of workspace
clear all;
close all;
clc;

%% Input parameters
X=(-1200:10:1200)'; % Range of x-coordinates for global grid [m]
Y=(-1000:10:1000)'; % Range of y-coordinates for global grid [m]
[x,y]=meshgrid(X,Y); % Construct global Cartesian grid (x,y) [m]
B=400; % Breakwater gap width [m]
x2=-0.5*B; y2=0; % Location of left breakwater head [m]
x1= 0.5*B; y1=0; % Location of right breakwater head [m]
wavedir_range=(-720:15:720); % Global wave direction range [degrees]
L=100; % Wave length [m]
R1=1.0; % Reflection coefficient of right breakwater [-]
R2=1.0; % Reflection coefficient of left breakwater [-]
BreakwaterLeft = 1; % Presence of breakwater on left side {0,1} [-]
BreakwaterRight = 1; % Presence of breakwater on right side {0,1} [-]
load Cornu.mat;

%% Convert degrees to radians
beta_range=wavedir_range*pi/180;

%% Loop over all imposed wave directions
uit=figure('Position',[200 50 1500 900]); % Initialize figure
d=sqrt((max(X)-min(X)).^2+(max(Y)-min(Y)).^2); % Initialize figure
n=length(beta_range); % Initialize maximum difference registration
maxdiff_Fi = zeros(n,1); % Initialize maximum difference registration
j=0; % Initialize counter
for beta=beta_range
    j=j+1; % Update counter
    % Left breakwater
    % Compute and store solution according to Penney and Price (1952)
    [F,Fi,Fr,Gi,Gr,Hi,Hr,sigma1,sigma2]=Sommerfeld_PenneyPrice(x,y,x2,y2,pi,beta,L,Cornu);
    Left.F = F; % Total wave field
    Left.Fi = Fi; % Incident wave field
    Left.Fr = Fr; % Reflected wave field
    Left.Gi = Gi; % Fresnel integral of the incident wave field
    Left.Gr = Gr; % Fresnel integral of the reflected wave field
    Left.Hi = Hi; % Undisturbed incident wave field
    Left.Hr = Hr; % Undisturbed reflected wave field
    Left.sigma1 = sigma1; % Integration boundaries for the incident wave field
    Left.sigma2 = sigma2; % Integration boundaries for the reflected wave field
    clear F Fi Fr Gi Gr Hi Hr sigma1 sigma2;
    % Compute phase shift in order to harmonize left and right solutions
    Left.I=exp(i*(0.5*B*cos(beta)/L)*2*pi);
    if BreakwaterLeft==0
        Left.Gi=x*0+1;
        Left.Gr=x*0;
    end
end

```

```

end;
% Right breakwater
% Compute and store solution according to Penney and Price (1952)
[F,Fi,Fr,Gi,Gr,Hi,Hr,sigma1,sigma2]=Sommerfeld_PenneyPrice(x,y,x1,y1,0,beta,L,Cornu);
Right.F = F; % Total wave field
Right.Fi = Fi; % Incident wave field
Right.Fr = Fr; % Reflected wave field
Right.Gi = Gi; % Fresnel integral of the incident wave field
Right.Gr = Gr; % Fresnel integral of the reflected wave field
Right.Hi = Hi; % Undisturbed incident wave field
Right.Hr = Hr; % Undisturbed reflected wave field
Right.sigma1 = sigma1; % Integration boundaries for the incident wave field
Right.sigma2 = sigma2; % Integration boundaries for the reflected wave field
clear F Fi Fr Gi Gr Hi Hr sigma1 sigma2;
% Compute phase shift in order to harmonize left and right solutions
Right.I=exp(i*(-0.5*B*cos(beta)/L)*2*pi);
if BreakwaterRight==0
    Right.Gi=x*0+1;
    Right.Gr=x*0;
end;
% Solution assembly
Right.Fi = Right.I.*Right.Hi.*(Right.Gi+Left.Gi-1);
Right.Fr = Right.I.*Right.Hr.*Right.Gr;
Left.Fi = Left.I.*Left.Hi.*(Right.Gi+Left.Gi-1);
Left.Fr = Left.I.*Left.Hr.*Left.Gr;
Fi = (Right.Fi+Left.Fi)*0.5;
Fr = R1*Right.Fr+R2*Left.Fr;
F = Fi+Fr;
% Compute largest difference between left and right solution of incident wave
maxdiff_Fi(j) = max(max(abs(Left.Fi-Right.Fi)));
% Write largest difference between left and right solution of incident wave to screen
disp(['Wave direction = ',num2str(wavedir_range(j)),'\n']);
disp('Largest difference between left and right solution of incident wave:');
disp(num2str(maxdiff_Fi(j)));
disp(' ');
% Plot results: update figure
subplot(3,4,1);
pcolor(x,y,abs(Left.Fi));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)],[y2 y2+d*sin(beta)],'--g','linewidth',2);hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)],[y1 y1+d*sin(beta)],'--g','linewidth',2);hold on; end;
if BreakwaterLeft;
plot([min(X) x2],[0 0],'k','linewidth',2);hold on; end;
if BreakwaterRight;
plot([x1 max(X)],[0 0],'k','linewidth',2);hold on; end;
title(['\beta = ',num2str(wavedir_range(j)),'\n'],'fontsize',12,'fontweight','b');
hold off;
xlabel('Amplitude','fontsize',12,'fontweight','b');
ylabel('Incident wave (left)','fontsize',12,'fontweight','b');
subplot(3,4,2);
pcolor(x,y,angle(Left.Fi));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)],[y2 y2+d*sin(beta)],'--g','linewidth',2);hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)],[y1 y1+d*sin(beta)],'--g','linewidth',2);hold on; end;
if BreakwaterLeft;
plot([min(X) x2],[0 0],'k','linewidth',2);hold on; end;
if BreakwaterRight;
plot([x1 max(X)],[0 0],'k','linewidth',2);hold on; end;
title(['\beta = ',num2str(wavedir_range(j)),'\n'],'fontsize',12,'fontweight','b');
hold off;
xlabel('Phase','fontsize',12,'fontweight','b');
ylabel('Incident wave (left)','fontsize',12,'fontweight','b');
subplot(3,4,3);
pcolor(x,y,abs(Right.Fi));
shading interp;colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)],[y2 y2+d*sin(beta)],'--g','linewidth',2);hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)],[y1 y1+d*sin(beta)],'--g','linewidth',2);hold on; end;
if BreakwaterLeft;
plot([min(X) x2],[0 0],'k','linewidth',2);hold on; end;
if BreakwaterRight;

```

```

plot([x1 max(X)], [0 0], 'k', 'linewidth', 2); hold on; end;
title(['\beta = ', num2str(wavedir_range(j)), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('Amplitude', 'fontsize', 12, 'fontweight', 'b');
ylabel('Incident wave (right)', 'fontsize', 12, 'fontweight', 'b');
subplot(3, 4, 4);
pcolor(x, y, angle(Right.Fi));
shading interp; colorbar; set(gca, 'dataaspectratio', [1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)], [y2 y2+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)], [y1 y1+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterLeft;
plot([min(X) x2], [0 0], 'k', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 max(X)], [0 0], 'k', 'linewidth', 2); hold on; end;
title(['\beta = ', num2str(wavedir_range(j)), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('Phase', 'fontsize', 12, 'fontweight', 'b');
ylabel('Incident wave (right)', 'fontsize', 12, 'fontweight', 'b');
subplot(3, 4, 5);
pcolor(x, y, abs(Left.Fr));
shading interp; colorbar; set(gca, 'dataaspectratio', [1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)], [y2 y2+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)], [y1 y1+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterLeft;
plot([min(X) x2], [0 0], 'k', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 max(X)], [0 0], 'k', 'linewidth', 2); hold on; end;
title(['\beta = ', num2str(wavedir_range(j)), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('Amplitude', 'fontsize', 12, 'fontweight', 'b');
ylabel('Reflected wave (left)', 'fontsize', 12, 'fontweight', 'b');
subplot(3, 4, 6);
pcolor(x, y, angle(Left.Fr));
shading interp; colorbar; set(gca, 'dataaspectratio', [1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)], [y2 y2+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)], [y1 y1+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterLeft;
plot([min(X) x2], [0 0], 'k', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 max(X)], [0 0], 'k', 'linewidth', 2); hold on; end;
title(['\beta = ', num2str(wavedir_range(j)), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('Phase', 'fontsize', 12, 'fontweight', 'b');
ylabel('Reflected wave (left)', 'fontsize', 12, 'fontweight', 'b');
subplot(3, 4, 7);
pcolor(x, y, abs(Right.Fr));
shading interp; colorbar; set(gca, 'dataaspectratio', [1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)], [y2 y2+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)], [y1 y1+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterLeft;
plot([min(X) x2], [0 0], 'k', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 max(X)], [0 0], 'k', 'linewidth', 2); hold on; end;
title(['\beta = ', num2str(wavedir_range(j)), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('Amplitude', 'fontsize', 12, 'fontweight', 'b');
ylabel('Reflected wave (right)', 'fontsize', 12, 'fontweight', 'b');
subplot(3, 4, 8);
pcolor(x, y, angle(Right.Fr));
shading interp; colorbar; set(gca, 'dataaspectratio', [1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)], [y2 y2+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)], [y1 y1+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterLeft;
plot([min(X) x2], [0 0], 'k', 'linewidth', 2); hold on; end;

```

```

    if BreakwaterRight;
    plot([x1 max(X)], [0 0], 'k', 'linewidth', 2); hold on; end;
    title(['\beta = ', num2str(wavedir_range(j)), '°'], 'fontsize', 12, 'fontweight', 'b');
    hold off;
    xlabel('Phase', 'fontsize', 12, 'fontweight', 'b');
    ylabel('Reflected wave (right)', 'fontsize', 12, 'fontweight', 'b');
subplot(3, 4, 9);
pcolor(x, y, abs(F));
shading interp; colorbar; set(gca, 'dataaspectratio', [1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)], [y2 y2+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)], [y1 y1+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterLeft;
plot([min(X) x2], [0 0], 'k', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 max(X)], [0 0], 'k', 'linewidth', 2); hold on; end;
title(['\beta = ', num2str(wavedir_range(j)), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('Amplitude', 'fontsize', 12, 'fontweight', 'b');
ylabel('Total wave', 'fontsize', 12, 'fontweight', 'b');
subplot(3, 4, 10);
pcolor(x, y, angle(F));
shading interp; colorbar; set(gca, 'dataaspectratio', [1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)], [y2 y2+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)], [y1 y1+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterLeft;
plot([min(X) x2], [0 0], 'k', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 max(X)], [0 0], 'k', 'linewidth', 2); hold on; end;
title(['\beta = ', num2str(wavedir_range(j)), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('Phase', 'fontsize', 12, 'fontweight', 'b');
ylabel('Total wave', 'fontsize', 12, 'fontweight', 'b');
subplot(3, 4, 11);
%pcolor(x, y, abs(Left.Fi-Right.Fi));
pcolor(x, y, abs(Left.I*Left.Hi-Right.I*Right.Hi));
shading interp; colorbar; set(gca, 'dataaspectratio', [1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)], [y2 y2+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)], [y1 y1+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterLeft;
plot([min(X) x2], [0 0], 'k', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 max(X)], [0 0], 'k', 'linewidth', 2); hold on; end;
title(['\beta = ', num2str(wavedir_range(j)), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('Difference left-right', 'fontsize', 12, 'fontweight', 'b');
ylabel('Total wave', 'fontsize', 12, 'fontweight', 'b');
subplot(3, 4, 12);
%pcolor(x, y, abs(Left.Fi-Right.Fi));
pcolor(x, y, abs(Left.I*Left.Hr-Right.I*Right.Hr));
shading interp; colorbar; set(gca, 'dataaspectratio', [1 1 1]);
hold on;
if BreakwaterLeft;
plot([x2 x2+d*cos(beta)], [y2 y2+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 x1+d*cos(beta)], [y1 y1+d*sin(beta)], '--g', 'linewidth', 2); hold on; end;
if BreakwaterLeft;
plot([min(X) x2], [0 0], 'k', 'linewidth', 2); hold on; end;
if BreakwaterRight;
plot([x1 max(X)], [0 0], 'k', 'linewidth', 2); hold on; end;
title(['\beta = ', num2str(wavedir_range(j)), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('Difference left-right', 'fontsize', 12, 'fontweight', 'b');
ylabel('Total wave', 'fontsize', 12, 'fontweight', 'b');
drawnow;
% End of plot
end;
% End of loop

%% End of Matlab script

```

## Appendix F: Matlab script BoundaryConditions\_Gap.m

```

% Script BoundaryConditions_Gap.m
%
% This script computes and plots the values of the wave function F and its
% derivative dF/dy along each breakwater in a configuration with two
% breakwaters and a gap in between. The user can vary the reflection
% coefficient of each breakwater, in order to examine to which extent the
% boundary conditions of the analytical Sommerfeld (1896) solution for 2D
% diffraction are satisfied. Because the values of F and dF/dy can be
% different on each side of a breakwater, both values are given; the one value
% applies to the "lee" side of the breakwater configuration, and the other
% value applies to the "exposed" side of the breakwaters.
% The script plots the absolute values of F and dF/dy as a function of x,
% for y = 0, which represents the line which coincides with both
% breakwaters. (Both the results for the lee side and the exposed side are
% plotted.)
%
% (c) H. Talstra, Svasek Hydraulics, Rotterdam, The Netherlands, 26 February 2014
%
% The script makes use of the following Matlab .mat-file:
%   Cornu.mat
%
% N.B. Within the context of this Matlab script, no information about
% wave height/amplitude, wave period and water depth is required. The
% resulting wave function F is valid for an incident wave with unit amplitude
% and the user may multiply it by the actual incident wave height/amplitude. The
% necessary information about wave period and water depth is already accounted
% for by the specified wave length/wave number (via the dispersion relation).

%% Cleanup of workspace
clear all;
close all;
clc;

%% Input parameters
x=(-1200:10:1200)';           % Range of x-coordinates along breakwater line [m]
wavedir_range=(-720:1:720);  % Global wave direction range [degrees]
B=400;                       % Breakwater gap width [m]
L=100;                       % Wave length [m]
R1=1.0;                      % Reflection coefficient of right breakwater [-]
R2=1.0;                      % Reflection coefficient of left breakwater [-]
load Cornu.mat;

%% Convert degrees to radians
beta_range=wavedir_range*pi/180;

%% Loop over all imposed wave directions
uit=figure('Position',[200 50 1500 900]); % Initialize figure
n=length(beta_range); % Initialize maximum registration
maxima = zeros(n,8); % Initialize maximum registration
j=0; % Initialize counter
for beta=beta_range
    j=j+1; % Update counter
    % Left breakwater
    x0 = -0.5*B;
    delta = pi;
    k = 2*pi/L;
    theta0 = beta-delta;
    xx = (x-x0)*cos(delta);
    r = sqrt(xx.*xx);
    r2 = r;
    jj2 = find(xx>=0);
    theta = r*0+pi; theta(jj2)=0;
    theta_2a = theta;
    sigma_i2a = 2*sqrt(k*r/pi).*sin(0.5*(theta-theta0)).*sign(cos(0.5*(theta0)));
    sigma_r2a = -2*sqrt(k*r/pi).*sin(0.5*(theta+theta0)).*sign(cos(0.5*(theta0)));
    tau_i2a = 2*sqrt(k*r/pi).*cos(0.5*(theta-theta0)).*sign(cos(0.5*(theta0)));
    tau_r2a = -2*sqrt(k*r/pi).*cos(0.5*(theta+theta0)).*sign(cos(0.5*(theta0)));
    H_i2a = exp(-i*k*r.*cos(theta-theta0));
    H_r2a = exp(-i*k*r.*cos(theta+theta0));
    theta = r*0+pi; theta(jj2)=2*pi;
    theta_2b = theta;
    sigma_i2b = 2*sqrt(k*r/pi).*sin(0.5*(theta-theta0)).*sign(cos(0.5*(theta0)));
    sigma_r2b = -2*sqrt(k*r/pi).*sin(0.5*(theta+theta0)).*sign(cos(0.5*(theta0)));
    tau_i2b = 2*sqrt(k*r/pi).*cos(0.5*(theta-theta0)).*sign(cos(0.5*(theta0)));
    tau_r2b = -2*sqrt(k*r/pi).*cos(0.5*(theta+theta0)).*sign(cos(0.5*(theta0)));
    G_i2a = 0.5+interp1(Cornu.sigma,Cornu.G,sigma_i2a);

```



```

G_r2a = 0.5+interp1(Cornu.sigma,Cornu.G,sigma_r2a);
G_i2b = 0.5+interp1(Cornu.sigma,Cornu.G,sigma_i2b);
G_r2b = 0.5+interp1(Cornu.sigma,Cornu.G,sigma_r2b);
H_i2b = exp(-i*k*r.*cos(theta-theta0));
H_r2b = exp(-i*k*r.*cos(theta+theta0));
I2 = exp(i*0.5*k*B*cos(beta));
% Right breakwater
x0 = 0.5*B;
delta = 0;
k = 2*pi/L;
theta0 = beta-delta;
xx = (x-x0)*cos(delta);
r = sqrt(xx.*xx);
r1 = r;
jj1 = find(xx>=0);
theta = r*0+pi; theta(jj1)=0;
theta_la = theta;
sigma_ila = 2*sqrt(k*r/pi).*sin(0.5*(theta-theta0)).*sign(cos(0.5*(theta0)));
sigma_rla = -2*sqrt(k*r/pi).*sin(0.5*(theta+theta0)).*sign(cos(0.5*(theta0)));
tau_ila = 2*sqrt(k*r/pi).*cos(0.5*(theta-theta0)).*sign(cos(0.5*(theta0)));
tau_rla = -2*sqrt(k*r/pi).*cos(0.5*(theta+theta0)).*sign(cos(0.5*(theta0)));
H_ila = exp(-i*k*r.*cos(theta-theta0));
H_rla = exp(-i*k*r.*cos(theta+theta0));
theta = r*0+pi; theta(jj1)=2*pi;
theta_lb = theta;
sigma_ilb = 2*sqrt(k*r/pi).*sin(0.5*(theta-theta0)).*sign(cos(0.5*(theta0)));
sigma_rlb = -2*sqrt(k*r/pi).*sin(0.5*(theta+theta0)).*sign(cos(0.5*(theta0)));
tau_ilb = 2*sqrt(k*r/pi).*cos(0.5*(theta-theta0)).*sign(cos(0.5*(theta0)));
tau_rlb = -2*sqrt(k*r/pi).*cos(0.5*(theta+theta0)).*sign(cos(0.5*(theta0)));
G_ila = 0.5+interp1(Cornu.sigma,Cornu.G,sigma_ila);
G_rla = 0.5+interp1(Cornu.sigma,Cornu.G,sigma_rla);
G_ilb = 0.5+interp1(Cornu.sigma,Cornu.G,sigma_ilb);
G_rlb = 0.5+interp1(Cornu.sigma,Cornu.G,sigma_rlb);
H_ilb = exp(-i*k*r.*cos(theta-theta0));
H_rlb = exp(-i*k*r.*cos(theta+theta0));
I1 = exp(-i*0.5*k*B*cos(beta));
% Solution assembly
F_ila = G_ila.*H_ila*I1; % Incident wave 1 (lee)
F_ilb = G_ilb.*H_ilb*I1; % Incident wave 1 (exposed)
F_rla = G_rla.*H_rla*I1; % Reflected wave 1 (lee)
F_rlb = G_rlb.*H_rlb*I1; % Reflected wave 1 (exposed)
F_la = G_ila.*H_ila*I1+R1.*G_rla.*H_rla*I1; % Total wave 1 (lee)
F_lb = G_ilb.*H_ilb*I1+R1.*G_rlb.*H_rlb*I1; % Total wave 1 (exposed)
F_i2a = G_i2a.*H_i2a*I2; % Incident wave 2 (exposed)
F_i2b = G_i2b.*H_i2b*I2; % Incident wave 2 (lee)
F_r2a = G_r2a.*H_r2a*I2; % Reflected wave 2 (exposed)
F_r2b = G_r2b.*H_r2b*I2; % Reflected wave 2 (lee)
F_2a = G_i2a.*H_i2a*I2+R2.*G_r2a.*H_r2a*I2; % Total wave 1 (exposed)
F_2b = G_i2b.*H_i2b*I2+R2.*G_r2b.*H_r2b*I2; % Total wave 1 (lee)
H_i0=exp(-i*k*x*cos(beta)); % Undisturbed incident wave
H_r0=exp(-i*k*x*cos(beta)); % Undisturbed reflected wave
F_a=F_la+F_2b-H_i0; % Total wave (lee)
F_b=F_lb+F_2a-H_i0; % Total wave (exposed)
% Assembly of spatial derivatives
dF1dy=(I1/2./r1).*((1+i)/2).*exp(-
i*k.*r1).*(tau_ila+R1*tau_rla)+((G_ila.*H_ila.*I1)./r1).*i*k.*r1.*sin(theta_la-beta)
+((G_rla.*H_rla.*I1)./r1).*R1.*i*k.*r1.*sin(theta_la+beta);
dF2dy=(I2/2./r2).*((1+i)/2).*exp(-
i*k.*r2).*(tau_i2a+R2*tau_r2a)+((G_i2a.*H_i2a.*I2)./r2).*i*k.*r2.*sin(theta_2a-
beta+pi)+((G_r2a.*H_r2a.*I2)./r2).*R2.*i*k.*r2.*sin(theta_2a+beta-pi);
dF3dy=H_i0.*i*k.*sin(beta);
dFdy_la=dF1dy+dF2dy+dF3dy; % Right breakwater (lee)
clear dF1dy dF2dy dF3dy;
dF1dy=(I1/2./r1).*((1+i)/2).*exp(-
i*k.*r1).*(tau_ilb+R1*tau_rlb)+((G_ilb.*H_ilb.*I1)./r1).*i*k.*r1.*sin(theta_lb-beta)
+((G_rlb.*H_rlb.*I1)./r1).*R1.*i*k.*r1.*sin(theta_lb+beta);
dF2dy=(I2/2./r2).*((1+i)/2).*exp(-
i*k.*r2).*(tau_i2b+R2*tau_r2b)+((G_i2b.*H_i2b.*I2)./r2).*i*k.*r2.*sin(theta_2b-
beta+pi)+((G_r2b.*H_r2b.*I2)./r2).*R2.*i*k.*r2.*sin(theta_2b+beta-pi);
dF3dy=H_i0.*i*k.*sin(beta);
dFdy_lb=dF1dy+dF2dy+dF3dy; % Right breakwater (exposed)
clear dF1dy dF2dy dF3dy;
dF1dy=-(I1/2./r1).*((1+i)/2).*exp(-i*k.*r1).*(tau_ila+R1*tau_rla)-
((G_ila.*H_ila.*I1)./r1).*i*k.*r1.*sin(theta_la-beta) -
((G_rla.*H_rla.*I1)./r1).*R1.*i*k.*r1.*sin(theta_la+beta);
dF2dy=-(I2/2./r2).*((1+i)/2).*exp(-i*k.*r2).*(tau_i2a+R2*tau_r2a)-
((G_i2a.*H_i2a.*I2)./r2).*i*k.*r2.*sin(theta_2a-beta+pi)-
((G_r2a.*H_r2a.*I2)./r2).*R2.*i*k.*r2.*sin(theta_2a+beta-pi);
dF3dy=H_i0.*i*k.*sin(beta);

```

```

dFdy_2a=dF1dy+dF2dy+dF3dy; % Left breakwater (exposed)
clear dF1dy dF2dy dF3dy;
dF1dy=-(I1/2./r1).*(1+i)/2).*exp(-i*k.*r1).*(tau_i1b+R1*tau_r1b)-
((G_i1b.*H_i1b.*I1)./r1).*i*k.*r1.*sin(theta_1b-beta) -
((G_r1b.*H_r1b.*I1)./r1).*R1*i*k.*r1.*sin(theta_1b+beta);
dF2dy=-(I2/2./r2).*(1+i)/2).*exp(-i*k.*r2).*(tau_i2b+R2*tau_r2b)-
((G_i2b.*H_i2b.*I2)./r2).*i*k.*r2.*sin(theta_2b-beta+pi)-
((G_r2b.*H_r2b.*I2)./r2).*R2*i*k.*r2.*sin(theta_2b+beta-pi);
dF3dy=H_i0*i*k*sin(beta);
dFdy_2b=dF1dy+dF2dy+dF3dy; % Left breakwater (lee)
clear dF1dy dF2dy dF3dy;
% Compute global maxima of absolute values of F and dF/dy,
% for both breakwaters (both lee side and exposed side)
maxima(j,:)=max(abs(F_a(jj1))) max(abs(F_b(jj1))) max(abs(F_a(jj2))) max(abs
(F_b(jj2))) max(abs(dFdy_1a(jj1))) max(abs(dFdy_1b(jj1))) max(abs(dFdy_2a(jj2))) max(abs
(dFdy_2b(jj2))));
% Plot results
subplot(2,1,1); % Total wave
plot(x(jj1),abs(F_a(jj1)),'g','linewidth',3);hold on; % Right breakwater (lee)
plot(x(jj1),abs(F_b(jj1)),'--b','linewidth',3);hold on; % Right breakwater
(exposed)
plot(x(jj2),abs(F_a(jj2)),'g','linewidth',3);hold on; % Left breakwater (lee)
plot(x(jj2),abs(F_b(jj2)),'--b','linewidth',3);hold on; % Left breakwater
(exposed)
title(['Boundary condition |F| for \beta = ',num2str(wavedir_range(j)),'°', R_1 =
',num2str(R1)', ' R_2 = ',num2str(R2)],'fontsize',12,'fontweight','b');
legend('At lee side of breakwater','At exposed side of breakwater');
ax=axis;ax(3)=-0.1*ax(4);axis(ax);
axis([min(x) max(x) -0.5 2.5]);
text(min(x)*0.85+0.15*max(x),-0.1,'|F| along left breakwater','fontsize',12);ho;
text(min(x)*0.35+0.65*max(x),-0.1,'|F| along right breakwater','fontsize',12);ho;
hold off;
subplot(2,1,2); % Spatial derivative
plot(x(jj1),abs(dFdy_1a(jj1)),'g','linewidth',3);hold on; % Right breakwater (lee)
plot(x(jj1),abs(dFdy_1b(jj1)),'--b','linewidth',3);hold on; % Right breakwater
(exposed)
plot(x(jj2),abs(dFdy_2b(jj2)),'g','linewidth',3);hold on; % Left breakwater (lee)
plot(x(jj2),abs(dFdy_2a(jj2)),'--b','linewidth',3);hold on; % Left breakwater
(exposed)
title(['Boundary condition |dF/dy| for \beta = ',num2str(wavedir_range(j)),'°', R_1 =
',num2str(R1)', ' R_2 = ',num2str(R2)],'fontsize',12,'fontweight','b');
legend('At lee side of breakwater','At exposed side of breakwater');
ax=axis;ax(3)=-0.1*ax(4);axis(ax);
axis([min(x) max(x) -0.05 0.25]);
text(min(x)*0.85+0.15*max(x),-0.01,'|dF/dy| along left breakwater','fontsize',12);ho;
text(min(x)*0.35+0.65*max(x),-0.01,'|dF/dy| along right breakwater','fontsize',12);ho;
hold off;
drawnow;
% End of plot
end;
% End of loop

%% End of Matlab script

```

## Appendix G: Matlab script SpectralIntegral.m

```

% Script SpectralIntegral.m
%
% This Matlab script builds and integrates the directional wave spectrum by
% Mitsayasu, as proposed by Goda (1978).
% In particular, it checks whether the resulting spectrum has the correct
% wave energy content.
%
% (c) H. Talstra, Svasek Hydraulics, Rotterdam, The Netherlands, 10 March 2014

%% Clean up workspace
clear all;
close all;
clc;

%% Input
Hsig=1; % Significant wave height H(1/3) [m]
Tsig=8; % Significant wave period T(1/3) [s]
Smax=10; % Wave directional spreading parameter (Goda, 1978) [-]
fmax=1; % Maximum wave frequency considered [Hz]
nwavefreq=100; % Number of wave frequencies considered [-]
nwavedir=200; % Number of wave directions considered [-]

%% Polar domain (frequencies and directions)
F=(0:(1/nwavefreq):1)*fmax; % Range of wave frequencies (Hz)
THETA=(0:(1/nwavedir):1)*2*pi-pi; % Range of wave directions (rad) (IMPORTANT: must range
from -pi to +pi, NOT from 0 to 2*pi!)
df=mean(diff(F)); % Increment of wave frequencies (Hz)
dth=mean(diff(THETA)); % Increment of wave directions (rad)
[f,theta]=meshgrid(F,THETA); % Make grid of polar coordinates
x=f.*cos(theta); % Make corresponding x-coordinate
y=f.*sin(theta); % Make corresponding y-coordinate
fp=1./(1.05.*Tsig); % Peak wave frequency (according to Goda, 1978) [Hz]

%% Auxiliary Cartesian domain (XY)
XX=(-1:(1/nwavefreq):1)*fmax; % Range of Cartesian x-coordinates
YY=(-1:(1/nwavedir):1)*fmax; % Range of Cartesian y-coordinates
dx=mean(diff(XX)); % Increment of x-coordinates
dy=mean(diff(YY)); % Increment of y-coordinates
[xx,yy]=meshgrid(XX,YY); % Make grid of Cartesian coordinates

%% Frequency spectrum (one-dimensional, no directional information)
% Mitsayasu frequency spectrum (according to Goda, 1978) [m^2.s]
Sf_1D=0.257*(Hsig^2)*(Tsig^-4)*(f.^-5).*exp(-1.03.*(Tsig.*f).^-4);
% Replace NaN's by zeros for locations with zero frequency
Sf_1D(isnan(Sf_1D))=0;
% Compute zero-th moment m0 of spectrum (spectral wave energy)
m0_1D=trapz(F,Sf_1D);
% Compute second moment m2 of spectrum
m2_1D=trapz(F,F.*f.*Sf_1D);
% Compute significant wave height Hm0 (should be equal to Hsig)
Hm0_1D=4.004*sqrt(m0_1D);
% Compute mean zero-crossing period T0 (is linearly proportional to Tsig)
T0_1D=sqrt(m0_1D./m2_1D);
% Spectral peak density for peak wave period
Sfp=0.257*(Hsig^2)*(Tsig^-4)*(fp.^-5).*exp(-1.03.*(Tsig.*fp).^-4);

%% Frequency spectrum (two-dimensional, polar domain)
% Mitsayasu frequency spectrum (according to Goda, 1978) [m^2.s] (N.B. divide by 2*pi !!!)
Sf=(0.257*(Hsig^2)*(Tsig^-4)*(f.^-5).*exp(-1.03.*(Tsig.*f).^-4))/(2*pi);
% Replace NaN's by zeros for locations with zero frequency
Sf(isnan(Sf))=0;
% Compute zero-th moment m0 of spectrum (spectral wave energy)
m0=trapz(THETA,(trapz(F,Sf,2)));
% Compute second moment m2 of spectrum
m2=trapz(THETA,(trapz(F,f.*f.*Sf,2)));
% Compute significant wave height Hm0 (should be equal to Hsig)
Hm0=4.004*sqrt(m0);
% Compute mean zero-crossing period T0 (is linearly proportional to Tsig)
T0=sqrt(m0./m2);

%% Frequency spectrum (two-dimensional, polar domain, divided by Jacobian)
% Mitsayasu frequency spectrum (Goda, 1978) [m^2.s] (N.B. divide by 2*pi and f !!!)
Sf_J=(0.257*(Hsig^2)*(Tsig^-4)*(f.^-5).*exp(-1.03.*(Tsig.*f).^-4))/(2*pi)./f;
% Replace NaN's by zeros for locations with zero frequency
Sf_J(isnan(Sf_J))=0;
% Compute zero-th moment m0 of spectrum (spectral wave energy) (N.B. multiply by f !!!)

```

```

m0_J=trapz(THETA,(trapz(F,Sf_J.*f,2)));
% Compute second moment m2 of spectrum (N.B. multiply by f !!!)
m2_J=trapz(THETA,(trapz(F,f.*f.*Sf_J.*f,2)));
% Compute significant wave height Hm0 (should be equal to Hsig)
Hm0_J=4.004*sqrt(m0_J);
% Compute mean zero-crossing period T0 (is linearly proportional to Tsig)
T0_J=sqrt(m0_J./m2_J);

%% Frequency spectrum (two-dimensional, Cartesian domain)
% Check: convert frequency spectrum Sf to XY-domain and integrate again...
Sf_xyJ=griddata(x,y,Sf_J,xx,yy);
% Replace NaN's by zeros (in order to fill corners of square domain)
Sf_xyJ(isnan(Sf_xyJ))=0;
% Compute zero-th moment m0 of spectrum (spectral wave energy)
m0_xyJ=trapz(YY,trapz(XX,(Sf_xyJ)));
% Compute second moment m2 of spectrum
m2_xyJ=trapz(YY,(trapz(XX,pyth(xx,yy).*pyth(xx,yy).*Sf_xyJ,2)));
% Compute significant wave height (should be equal to Hsig...)
Hm0_xyJ=4.004*sqrt(m0_xyJ);
% Compute mean zero-crossing period T0 (is linearly proportional to Tsig)
T0_xyJ=sqrt(m0_xyJ./m2_xyJ);

%% Directional spreading function (polar domain)
% Define power-law function S, based on parameter Smax
% Smax = 10 for wind waves, Smax = 75 for swell (according to Goda, 1978)
S=Smax*(f./fp).^5;
j=find(f>=fp);
S(j)=Smax*(f(j)./fp).^~2.5;
% Define spreading function G
G=cos(0.5*theta).^(2.*S);
% Normalize spreading function G by a factor G0 (for each frequency separately!)
G0=ones(201,1)*trapz(THETA,G)/(2*pi);
H=G./G0;
H0=ones(201,1)*trapz(THETA,H)/(2*pi);
% Attention: in this Matlab script, H is the normalized G function as
% defined by Goda (1978).

%% Compute final spectrum (two-dimensional, polar domain)
% Final spectrum; is equal to Sf multiplied by normalized spreading function H
E=Sf.*H;
E(isnan(E))=0;
% Compute zero-th moment m0 of spectrum (spectral wave energy)
m0_E=trapz(THETA,(trapz(F,E,2)));
% Compute second moment m2 of spectrum
m2_E=trapz(THETA,(trapz(F,f.*f.*E,2)));
% Compute significant wave height Hm0 (should be equal to Hsig)
Hm0_E=4.004*sqrt(m0_E);
% Compute mean zero-crossing period T0 (is linearly proportional to Tsig)
T0_E=sqrt(m0_E./m2_E);
% Integrate final spectrum over directions only
% (should be equal to the original frequency spectrum Sf!)
Sf_1D_check=trapz(THETA,E);
Sf_check=ones(201,1)*trapz(THETA,E)/(2*pi);

%% Compute final spectrum (two-dimensional, polar domain, divided by Jacobian)
% Final spectrum; is equal to Sf_J multiplied by normalized spreading function H
EJ=Sf_J.*H;
EJ(isnan(EJ))=0;
% Compute zero-th moment m0 of spectrum (spectral wave energy) (N.B. multiply by f !!!)
m0_EJ=trapz(THETA,(trapz(F,EJ.*f,2)));
% Compute second moment m2 of spectrum (N.B. multiply by f !!!)
m2_EJ=trapz(THETA,(trapz(F,f.*f.*EJ.*f,2)));
% Compute significant wave height Hm0 (should be equal to Hsig)
Hm0_EJ=4.004*sqrt(m0_EJ);
% Compute mean zero-crossing period T0 (is linearly proportional to Tsig)
T0_EJ=sqrt(m0_EJ./m2_EJ);
% Integrate final spectrum over directions only
% (should be equal to the original frequency spectrum Sf!)
Sf_1D_check2=trapz(THETA,EJ.*f);
Sf_check2=ones(201,1)*trapz(THETA,EJ.*f)/(2*pi);

%% Compute final spectrum (two-dimensional, Cartesian domain)
% Check: convert final spectrum EJ to XY-domain and integrate again...
Sf_xyEJ=griddata(x,y,EJ,xx,yy);
Sf_xyEJ(isnan(Sf_xyEJ))=0;
% Compute zero-th moment m0 of spectrum (spectral wave energy)
m0_xyEJ=trapz(YY,trapz(XX,(Sf_xyEJ)));
% Compute second moment m2 of spectrum
m2_xyEJ=trapz(YY,(trapz(XX,pyth(xx,yy).*pyth(xx,yy).*Sf_xyEJ,2)));

```

```

% Compute significant wave height Hm0 (should be equal to Hsig)
Hm0_xyEJ=4.004*sqrt(m0_xyEJ);
% Compute mean zero-crossing period T0 (is linearly proportional to Tsig)
T0_xyEJ=sqrt(m0_xyEJ./m2_xyEJ);

%% Plot some results
figure(1);
subplot(2,1,1);
plot(f',Sf','-');hold on;
plot(fp',Sfp/(2*pi)', 'or');hold on;
xlabel('f');ylabel('Sf');
subplot(2,1,2);
plot(f',S');
xlabel('f');ylabel('S');

figure(2);
subplot(2,2,1);
pcolor(x,y,f);shading interp;colorbar;axis equal;
title('f');xlabel('x=f*cos(theta)');ylabel('y=f*sin(theta)');
subplot(2,2,2);
pcolor(x,y,theta);shading interp;colorbar;axis equal;
title('theta');xlabel('x=f*cos(theta)');ylabel('y=f*sin(theta)');
subplot(2,2,3);
pcolor(x,y,Sf);shading interp;colorbar;axis equal;
title('Sf');xlabel('x=f*cos(theta)');ylabel('y=f*sin(theta)');
subplot(2,2,4);
pcolor(x,y,S);shading interp;colorbar;axis equal;
title('S');xlabel('x=f*cos(theta)');ylabel('y=f*sin(theta)');

figure(3);
subplot(2,2,1);
pcolor(x,y,G);shading interp;colorbar;axis equal;
title('G');xlabel('x=f*cos(theta)');ylabel('y=f*sin(theta)');
subplot(2,2,2);
pcolor(x,y,H);shading interp;colorbar;axis equal;
title('H');xlabel('x=f*cos(theta)');ylabel('y=f*sin(theta)');
subplot(2,2,3);
pcolor(x,y,E);shading interp;colorbar;axis equal;
title('E');xlabel('x=f*cos(theta)');ylabel('y=f*sin(theta)');
subplot(2,2,4);
pcolor(x,y,Sf_check);shading interp;colorbar;axis equal;
title('Sf_check','interpreter','none');xlabel('x=f*cos(theta)');ylabel('y=f*sin(theta)');

%% End of Matlab script

```

## Appendix H: Matlab function WaveNumber.m

```

function k=WaveNumber(D,T)
% function k=wave_number(D,T)
%
% This function computes the wave number as a function of water depth and wave period,
% using the standard dispersion relation for free-surface gravity waves. As the wave
% period is known here, the wave number must be determined iteratively (using a simple
% Newton-Raphson iteration); in contrast, if the wave number is known, the wave period
% could be derived directly.
%
% (c) H. Talstra, Svasek Hydraulics, Rotterdam, The Netherlands, 10 March 2014

g=9.81; % Gravity acceleration
omega=2*pi/T; % Angular frequency
k=omega*omega/g; % Deep-water limit of wave number
for iter=1:100 % Maximum of 100 iteration steps
    kk=k; % i-th iterate of wave number
    kD=kk*D; % Dimensionless wave number
    omegal=sqrt(g*kk*tanh(kD)); % Update of angular frequency
    n1=0.5 + kD/sinh(2*kD); % Compute n-factor for waves
    cg=n1*omegal/kk; % Group velocity
    dk=(omegal-omega)/cg; % Wave number increment (Newton-Raphson iteration)
    k=k-dk; % Wave number update
    if (abs(dk)<1.E-8); continue; end; % Repeat until converged
end;

%% End of Matlab function

```

## Appendix I: Matlab script GodaDiagram\_SemiInfiniteBreakwater.m

```

% Script GodaDiagram_SemiInfiniteBreakwater.m
%
% This script computes and plots the 2D diffraction diagram of waves
% around a semi-infinite breakwater at constant water depth. Frequency
% spreading and directional spreading are incorporated into the solution
% following the approach by Goda et al. (1978), whereas the analytical
% Sommerfeld (1896) solution for 2D diffraction around a semi-infinite breakwater
% is used in order to compute the elementary solution for each wave frequency and
% direction. The user may specify the computational domain, the water depth,
% the breakwater location and orientation, dominant wave length, dominant wave direction
% and the breakwater reflection coefficient.
% The effective diffraction coefficient (i.e. the incident/diffracted wave
% height ratio) "K_d_eff" and the effective incident/diffracted wave period
% ratio "K_d_T" are plotted into a figure (as a function of 2D position).
% The user is free (but obviously recommended) to write the results to a
% file after performing this Matlab script.
%
% (c) H. Talstra, Svasek Hydraulics, Rotterdam, The Netherlands, 10 July 2014
%
% The script makes use of the following Matlab .mat-file:
%   Cornu.mat
% and the following Matlab function:
%   WaveNumber.m
%   Sommerfeld_PenneyPrice.m
%
% N.B. The user has to specify a constant water depth, which determines the
% dispersion relation of free-surface gravity waves. This dispersion
% relation is needed in order to relate the spectral spreading of wave
% periods to the corresponding spectral spreading of wave numbers and wave
% lengths.

%% Clean up workspace
clear all;
close all;
clc;

%% Geometrical input parameters
X=(-1000:100:1000)'; % Range of x-coordinates for global grid [m]
Y=(0:100:2000)'; % Range of y-coordinates for global grid [m]
x0=0; % Global x-coordinates of the breakwater head [m]
y0=0; % Global y-coordinates of the breakwater head [m]
breakwaterdir=0; % Orientation of the breakwater
[degrees]
depth=1000000; % Water depth [m]
R=1.0; % Reflection coefficient breakwater [-]
load Cornu.mat;
afkappen = 1;

%% Spectral input parameters
wavedir=90; % Dominant global wave direction
[degrees]
wavelength=100000; % Representative wave length of incident wave [m]
Hsig=1; % Significant wave height H(1/3) of incident wave [m]
Smax=10; % Directional spreading parameter (Goda, 1978) [-]
nwavefreq=100; % Number of wave frequencies considered [-]
nwavedir=200; % Number of wave directions considered [-]

%% Convert degrees to radians
delta=breakwaterdir*pi/180; % Orientation of the breakwater in radians [rad]
beta0=wavedir*pi/180; % Dominant global wave direction in radians [rad]

%% Convert representative wave length to significant wave period
k=2*pi/wavelength; % Representative wave number [rad/m]
omega=sqrt(9.81*k*tanh(k*depth)); % Angular frequency [rad/s]
Tsig=2*pi/omega; % Significant wave period T(1/3) of incident wave [s]
fmax=10/Tsig; % Maximum wave frequency considered [Hz]

%% Setup of geometrical Cartesian domain (x and y)
[x,y]=meshgrid(X,Y); % Construct global Cartesian grid (x,y) [m]

%% Setup of spectral polar domain (frequencies and directions)
F=((1/nwavefreq):(1/nwavefreq):1)*fmax; % Range of wave frequencies (Hz)
THETA=(0:(1/nwavedir):1)*2*pi-pi; % Range of wave directions (rad) (IMPORTANT: must range
from -pi to +pi, NOT from 0 to 2*pi!)
df=mean(diff(F)); % Increment of wave frequencies (Hz)
dth=mean(diff(THETA)); % Increment of wave directions (rad)

```

```

[f,theta]=meshgrid(F,THETA);      % Make grid of polar coordinates
fp=1./(1.05.*Tsig);               % Peak wave frequency (according to Goda, 1978) [Hz]

%% Frequency spectrum (two-dimensional, polar domain)
% Mitsayasu frequency spectrum (according to Goda, 1978) [m^2·s] (N.B. divide by 2*pi !!!)
Sf=(0.257*(Hsig^2)*(Tsig^-4)*(f.^-5).*exp(-1.03.*(Tsig.*f).^-4))/(2*pi);
% Replace NaN's by zeros for locations with zero frequency
Sf(isnan(Sf))=0;

%% Directional spreading function (polar domain)
% Define power-law function S, based on parameter Smax
% Smax = 10 for wind waves, Smax = 75 for swell (according to Goda, 1978)
S=Smax*(f./fp).^5;
j=find(f>=fp);
S(j)=Smax*(f(j)./fp).^-2.5;
% Define spreading function D
D=cos(0.5*theta).^2.*S;
if afkappen
for j1=1:size(f,1)
    for j2=1:size(f,2)
        if cos(theta(j1,j2))<=0
            D(j1,j2)=0;
        end;
    end;
end;
if cos(beta0-delta)<0
if sin(beta0-delta)>0
    for j1=1:size(f,1)
        for j2=1:size(f,2)
            if sin(beta0+theta(j1,j2)-delta)<=0
                D(j1,j2)=0;
            end;
        end;
    end;
else
    for j1=1:size(f,1)
        for j2=1:size(f,2)
            if sin(beta0+theta(j1,j2)-delta)>0
                D(j1,j2)=0;
            end;
        end;
    end;
end;
end;
end;
% Normalize spreading function D by a factor D0 (for each frequency separately!)
D0=ones(nwavedir+1,1)*trapz(THETA,D)/(2*pi);
D=D./D0;
D2=ones(nwavedir+1,1)*trapz(THETA,D)/(2*pi);
clear D0 S j;

%% Compute final spectrum of incident wave (two-dimensional, polar domain)
% Final spectrum; is equal to frequency spectrum Sf multiplied by spreading function D
E_i=Sf.*D;
% Compute zero-th moment m0 of incident wave spectrum (spectral wave energy)
m0_i=sum(sum(E_i))*df*dth;
% Compute second moment m2 of incident wave spectrum
m2_i=sum(sum(f.*f.*E_i))*df*dth;
% Compute significant wave height Hm0 of incident wave (should be equal to Hsig)
Hm0_i=4.004*sqrt(m0_i);
% Compute mean zero-crossing period T0 of incident wave (is linearly proportional to Tsig)
T0_i=sqrt(m0_i./m2_i);
% Initialize zero-th moment m0 of diffracted wave spectrum
m0_d=x*0;
m0_da=x*0;
% Initialize second moment m2 of diffracted wave spectrum
m2_d=x*0;
m2_da=x*0;
clear Sf D;

tic;
%% Loop over all spectral components
for j1=1:size(f,1)
    for j2=1:size(f,2)
        ff=f(j1,j2);
        th=theta(j1,j2);
        beta=beta0+th;
        L=2*pi/WaveNumber(depth,1/ff);
        % Left breakwater

```



```

% Compute and store solution according to Penney and Price (1952)
[F,Fi,Fr,Gi,Gr,Hi,Hr,sigma1,sigma2]=Sommerfeld_PenneyPrice(x,y,x0,y0,delta,beta,L,Cornu);
% Solution assembly
F = Fi+R*Fr;
% Fa is a reference solution, which would occur if no breakwater
% were present. This reference solution will be used to check whether
% the Goda method is implemented correctly.
Fa=F*0+1;
% Compute spectral wave characteristics
% Spectral component of diffracted wave (for all locations x and y)
E_d=E_i(j1,j2).*abs(F).*abs(F);
% Spectral component of the reference solution (for all locations x and y)
E_da=E_i(j1,j2).*abs(Fa).*abs(Fa);
% Cumulative sum of zero-th moment m0 of diffracted wave spectrum
m0_d=m0_d+E_d*df*dth;
% Cumulative sum of zero-th moment m0 of the reference solution
m0_da=m0_da+E_da*df*dth;
% Cumulative sum of second moment m2 of diffracted wave spectrum
m2_d=m2_d+E_d*ff*ff*df*dth;
% Cumulative sum of second moment m2 of the reference solution
m2_da=m2_da+E_da*ff*ff*df*dth;
end;
ss=toc;
disp(['j1 = ',num2str(j1),', j2 = ',num2str(j2),', Time elapsed = ',num2str(ss),
sec.']);
end;
%% End of loop
tt=toc;
disp(['Time elapsed = ',num2str(tt), 'sec.']);

% Compute significant wave height Hm0 of diffracted wave
Hm0_d=4.004*sqrt(m0_d);
% Compute significant wave height Hm0 of the reference solution (should be equal to Hsig)
Hm0_da=4.004*sqrt(m0_da);
% Compute mean zero-crossing period T0 of diffracted wave (is linearly proportional to
Tsig)
T0_d=sqrt(m0_d./m2_d);
% Compute mean zero-crossing period T0 of the reference solution
T0_da=sqrt(m0_da./m2_da);

% Compute effective amplitude diffraction coefficient
K_d_eff = Hm0_d/Hm0_i;
% Compute effective amplitude coefficient of the reference solution
% (should be equal to 1)
K_d_effa = Hm0_da/Hm0_i;
% Compute effective wave period ratio of diffracted wave
K_d_T = T0_d/T0_i;
% Compute effective wave period ratio of the reference solution
% (should be equal to 1)
K_d_Ta = T0_da/T0_i;
clear K_d_Ta K_d_effa Hm0_da E_da T0_da m0_da m2_da;

%% Plot results to figure
uit=figure('Position',[200 50 1500 900]); % Initialize figure
d=sqrt((max(X)-min(X)).^2+(max(Y)-min(Y)).^2); % Initialize figure
subplot(1,2,1);
pcolor(x,y,K_d_eff);
shading interp; colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta0)], [y0 y0+d*sin(beta0)], '--g', 'linewidth', 2); hold on;
plot([x0 x0+d*cos(delta)], [y0 y0+d*sin(delta)], '-k', 'linewidth', 2); hold on;
title(['Effective wave amplitude ratio, \beta_0 =
',num2str(wavedir), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('x [m]', 'fontsize', 12, 'fontweight', 'b');
ylabel('y [m]', 'fontsize', 12, 'fontweight', 'b');
subplot(1,2,2);
pcolor(x,y,K_d_T);
shading interp; colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
plot([x0 x0+d*cos(beta0)], [y0 y0+d*sin(beta0)], '--g', 'linewidth', 2); hold on;
plot([x0 x0+d*cos(delta)], [y0 y0+d*sin(delta)], '-k', 'linewidth', 2); hold on;
title(['Effective wave period ratio, \beta_0 =
',num2str(wavedir), '°'], 'fontsize', 12, 'fontweight', 'b');
hold off;
xlabel('x [m]', 'fontsize', 12, 'fontweight', 'b');
ylabel('y [m]', 'fontsize', 12, 'fontweight', 'b');
%% End of Matlab script

```

## Appendix J: Matlab script GodaDiagram\_BreakwaterGap.m

```

% Script GodaDiagram_BreakwaterGap.m
%
% This script computes and plots the 2D diffraction diagram of waves around
% a gap in between two breakwaters at constant water depth. Frequency
% spreading and directional spreading are incorporated into the solution
% following the approach by Goda et al. (1978), whereas the analytical
% Sommerfeld (1896) solution for 2D diffraction around a semi-infinite breakwater
% is used in order to compute the elementary solution for each wave frequency and
% direction. The user may specify the computational domain, the water depth,
% the breakwater gap width, dominant wave length, dominant wave direction
% and two reflection coefficients (one for each breakwater).
% The effective diffraction coefficient (i.e. the incident/diffracted wave
% height ratio) "K_d_eff" and the effective incident/diffracted wave period
% ratio "K_d_T" are plotted into a figure (as a function of 2D position).
% The user is free (but obviously recommended) to write the results to a
% file after performing this Matlab script.
% Both of the breakwaters are optional and can be switched off on behalf of
% an easy comparison with solutions for the semi-infinite breakwater case.
%
% (c) H. Talstra, Svasek Hydraulics, Rotterdam, The Netherlands, 10 July 2014
%
% The script makes use of the following Matlab .mat-file:
%   Cornu.mat
% and the following Matlab function:
%   WaveNumber.m
%   Sommerfeld_PenneyPrice.m
%
% N.B. The user has to specify a constant water depth, which determines the
% dispersion relation of free-surface gravity waves. This dispersion
% relation is needed in order to relate the spectral spreading of wave
% periods to the corresponding spectral spreading of wave numbers and wave
% lengths.

%% Clean up workspace
clear all;
close all;
clc;

%% Geometrical input parameters
X=(0:20:300)'; % Range of x-coordinates for global grid [m]
Y=(0:20:400)'; % Range of y-coordinates for global grid [m]
B=100; % Breakwater gap width [m]
depth=1000000; % Water depth [m]
R1=1.0; % Reflection coefficient of right breakwater [-]
R2=1.0; % Reflection coefficient of left breakwater [-]
BreakwaterLeft = 1; % Presence of breakwater on left side {0,1} [-]
BreakwaterRight = 1; % Presence of breakwater on right side {0,1} [-]
load Cornu.mat;
afkappen = 1;

%% Spectral input parameters
wavedir=90; % Dominant global wave direction [degrees]
wavelength=100; % Representative wave length of incident wave [m]
Hsig=1; % Significant wave height H(1/3) of incident wave [m]
Smax=10; % Directional spreading parameter (Goda, 1978) [-]
nwavfreq=100; % Number of wave frequencies considered [-]
nwavedir=200; % Number of wave directions considered [-]

%% Convert degrees to radians
beta0=wavedir*pi/180; % Dominant global wave direction in radians [rad]

%% Convert representative wave length to significant wave period
k=2*pi/wavelength; % Representative wave number [rad/m]
omega=sqrt(9.81*k*tanh(k*depth)); % Angular frequency [rad/s]
Tsig=2*pi/omega; % Significant wave period T(1/3) of incident wave [s]
fmax=10/Tsig; % Maximum wave frequency considered [Hz]

%% Setup of geometrical Cartesian domain (x and y)
[x,y]=meshgrid(X,Y); % Construct global Cartesian grid (x,y) [m]
x2=-0.5*B; y2=0; % Location of left breakwater head [m]
x1= 0.5*B; y1=0; % Location of right breakwater head [m]

%% Setup of spectral polar domain (frequencies and directions)
F=((1/nwavfreq):(1/nwavfreq):1)*fmax; % Range of wave frequencies (Hz)

```

```

THETA=(0:(1/nwavedir):1)*2*pi-pi; % Range of wave directions (rad) (IMPORTANT: must range
from -pi to +pi, NOT from 0 to 2*pi!)
df=mean(diff(F)); % Increment of wave frequencies (Hz)
dth=mean(diff(THETA)); % Increment of wave directions (rad)
[f,theta]=meshgrid(F,THETA); % Make grid of polar coordinates
fp=1./(1.05.*Tsig); % Peak wave frequency (according to Goda, 1978) [Hz]

%% Frequency spectrum (two-dimensional, polar domain)
% Mitsayasu frequency spectrum (according to Goda, 1978) [m^2.s] (N.B. divide by 2*pi !!!)
Sf=(0.257*(Hsig^2)*(Tsig^-4)*(f.^-5).*exp(-1.03.*(Tsig.*f).^-4))/(2*pi);
% Replace NaN's by zeros for locations with zero frequency
Sf(isnan(Sf))=0;

%% Directional spreading function (polar domain)
% Define power-law function S, based on parameter Smax
% Smax = 10 for wind waves, Smax = 75 for swell (according to Goda, 1978)
S=Smax*(f./fp).^5;
j=find(f>=fp);
S(j)=Smax*(f(j)./fp).^-2.5;
% Define spreading function D
D=cos(0.5*theta).^2.*S;
if afkappen
if sin(beta0)>0
for j1=1:size(f,1)
for j2=1:size(f,2)
if sin(beta0+theta(j1,j2))<=0.1;
D(j1,j2)=0;
end;
end;
else
for j1=1:size(f,1)
for j2=1:size(f,2)
if sin(beta0+theta(j1,j2))>=-0.1;
D(j1,j2)=0;
end;
end;
end;
end;
% Normalize spreading function D by a factor D0 (for each frequency separately!)
D0=ones(nwavedir+1,1)*trapz(THETA,D)/(2*pi);
D=D./D0;
D2=ones(nwavedir+1,1)*trapz(THETA,D)/(2*pi);
clear D0 S j;

%% Compute final spectrum of incident wave (two-dimensional, polar domain)
% Final spectrum; is equal to frequency spectrum Sf multiplied by spreading function D
E_i=Sf.*D;
% Compute zero-th moment m0 of incident wave spectrum (spectral wave energy)
m0_i=sum(sum(E_i))*df*dth;
% Compute second moment m2 of incident wave spectrum
m2_i=sum(sum(f.*f.*E_i))*df*dth;
% Compute significant wave height Hm0 of incident wave (should be equal to Hsig)
Hm0_i=4.004*sqrt(m0_i);
% Compute mean zero-crossing period T0 of incident wave (is linearly proportional to Tsig)
T0_i=sqrt(m0_i./m2_i);
% Initialize zero-th moment m0 of diffracted wave spectrum
m0_d=x*0;
m0_da=x*0;
% Initialize second moment m2 of diffracted wave spectrum
m2_d=x*0;
m2_da=x*0;
clear Sf D;

tic;
%% Loop over all spectral components
for j1=1:size(f,1)
for j2=1:size(f,2)
ff=f(j1,j2);
th=theta(j1,j2);
beta=beta0+th;
L=2*pi/WaveNumber(depth,1/ff);
% Left breakwater
% Compute and store solution according to Penney and Price (1952)

[F,Fi,Fr,Gi,Gr,Hi,Hr,sigma1,sigma2]=Sommerfeld_PenneyPrice(x,y,x2,y2,pi,beta,L,Cornu);
Left.F = F; % Total wave field
Left.Fi = Fi; % Incident wave field

```

```

Left.Fr      = Fr;      % Reflected wave field
Left.Gi      = Gi;      % Fresnel integral of the incident wave field
Left.Gr      = Gr;      % Fresnel integral of the reflected wave field
Left.Hi      = Hi;      % Undisturbed incident wave field
Left.Hr      = Hr;      % Undisturbed reflected wave field
Left.sigma1  = sigma1;  % Integration boundaries for the incident wave field
Left.sigma2  = sigma2;  % Integration boundaries for the reflected wave field
clear F Fi Fr Gi Gr Hi Hr sigma1 sigma2;
% Compute phase shift in order to harmonize left and right solutions
Left.I=exp(i*(0.5*B*cos(beta)/L)*2*pi);
if BreakwaterLeft==0
    Left.Gi=x*0+1;
    Left.Gr=x*0;
end;
% Right breakwater
% Compute and store solution according to Penney and Price (1952)

[F,Fi,Fr,Gi,Gr,Hi,Hr,sigma1,sigma2]=Sommerfeld_PenneyPrice(x,y,x1,y1,0,beta,L,Cornu);
Right.F      = F;      % Total wave field
Right.Fi     = Fi;     % Incident wave field
Right.Fr     = Fr;     % Reflected wave field
Right.Gi     = Gi;     % Fresnel integral of the incident wave field
Right.Gr     = Gr;     % Fresnel integral of the reflected wave field
Right.Hi     = Hi;     % Undisturbed incident wave field
Right.Hr     = Hr;     % Undisturbed reflected wave field
Right.sigma1 = sigma1; % Integration boundaries for the incident wave field
Right.sigma2 = sigma2; % Integration boundaries for the reflected wave field
clear F Fi Fr Gi Gr Hi Hr sigma1 sigma2;
% Compute phase shift in order to harmonize left and right solutions
Right.I=exp(i*(-0.5*B*cos(beta)/L)*2*pi);
if BreakwaterRight==0
    Right.Gi=x*0+1;
    Right.Gr=x*0;
end;
% Solution assembly
Right.Fi = Right.I.*Right.Hi.*(Right.Gi+Left.Gi-1);
Right.Fr = Right.I.*Right.Hr.*Right.Gr;
Left.Fi  = Left.I.*Left.Hi.*(Right.Gi+Left.Gi-1);
Left.Fr  = Left.I.*Left.Hr.*Left.Gr;
Fi       = (Right.Fi+Left.Fi)*0.5;
Fr       = R1*Right.Fr+R2*Left.Fr;
F        = Fi+Fr;
% Fa is a reference solution, which would occur if no breakwaters
% were present. This reference solution will be used to check whether
% the Goda method is implemented correctly.
Fa=F*0+1;
% Compute spectral wave characteristics
% Spectral component of diffracted wave (for all locations x and y)
E_d=E_i(j1,j2).*abs(F).*abs(F);
% Spectral component of the reference solution (for all locations x and y)
E_da=E_i(j1,j2).*abs(Fa).*abs(Fa);
% Cumulative sum of zero-th moment m0 of diffracted wave spectrum
m0_d=m0_d+E_d*df*dth;
% Cumulative sum of zero-th moment m0 of the reference solution
m0_da=m0_da+E_da*df*dth;
% Cumulative sum of second moment m2 of diffracted wave spectrum
m2_d=m2_d+E_d*ff*ff*df*dth;
% Cumulative sum of second moment m2 of the reference solution
m2_da=m2_da+E_da*ff*ff*df*dth;
end;
ss=toc;
disp(['j1 = ',num2str(j1),' , j2 = ',num2str(j2),' , Time elapsed = ',num2str(ss),'
sec.']);
end;
%% End of loop
tt=toc;
disp(['Time elapsed = ',num2str(tt),'sec.']);

% Compute significant wave height Hm0 of diffracted wave
Hm0_d=4.004*sqrt(m0_d);
% Compute significant wave height Hm0 of the reference solution (should be equal to Hsig)
Hm0_da=4.004*sqrt(m0_da);
% Compute mean zero-crossing period T0 of diffracted wave (is linearly proportional to
Tsig)
T0_d=sqrt(m0_d./m2_d);
% Compute mean zero-crossing period T0 of the reference solution
T0_da=sqrt(m0_da./m2_da);

% Compute effective amplitude diffraction coefficient

```

```

K_d_eff = Hm0_d/Hm0_i;
% Compute effective amplitude coefficient of the reference solution
% (should be equal to 1)
K_d_effa = Hm0_da/Hm0_i;
% Compute effective wave period ratio of diffracted wave
K_d_T = T0_d/T0_i;
% Compute effective wave period ratio of the reference solution
% (should be equal to 1)
K_d_Ta = T0_da/T0_i;
clear K_d_Ta K_d_effa Hm0_da E_da T0_da m0_da m2_da;

%% Plot results to figure
uit=figure('Position',[200 50 1500 900]); % Initialize figure
d=sqrt((max(X)-min(X)).^2+(max(Y)-min(Y)).^2); % Initialize figure
subplot(1,2,1);
pcolor(x,y,K_d_eff);
shading interp; colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
if BreakwaterLeft; plot([x2 x2+d*cos(beta0)],[y2 y2+d*sin(beta0)],'--
g','linewidth',2);hold on; end;
if BreakwaterRight; plot([x1 x1+d*cos(beta0)],[y1 y1+d*sin(beta0)],'--
g','linewidth',2);hold on; end;
if BreakwaterLeft; plot([min(X) x2],[0 0],'k','linewidth',2);hold on; end;
if BreakwaterRight; plot([x1 max(X)],[0 0],'k','linewidth',2);hold on; end;
title(['Effective wave amplitude ratio, \beta_0 =
',num2str(wavedir),'°'],'fontsize',12,'fontweight','b');
hold off;
xlabel('x [m]','fontsize',12,'fontweight','b');
ylabel('y [m]','fontsize',12,'fontweight','b');
subplot(1,2,2);
pcolor(x,y,K_d_T);
shading interp; colorbar;set(gca,'dataaspectratio',[1 1 1]);
hold on;
if BreakwaterLeft; plot([x2 x2+d*cos(beta0)],[y2 y2+d*sin(beta0)],'--
g','linewidth',2);hold on; end;
if BreakwaterRight; plot([x1 x1+d*cos(beta0)],[y1 y1+d*sin(beta0)],'--
g','linewidth',2);hold on; end;
if BreakwaterLeft; plot([min(X) x2],[0 0],'k','linewidth',2);hold on; end;
if BreakwaterRight; plot([x1 max(X)],[0 0],'k','linewidth',2);hold on; end;
title(['Effective wave period ratio, \beta_0 =
',num2str(wavedir),'°'],'fontsize',12,'fontweight','b');
hold off;
xlabel('x [m]','fontsize',12,'fontweight','b');
ylabel('y [m]','fontsize',12,'fontweight','b');

%% End of Matlab script

```

**SVASEK**  
**HYDRAULICS**  
COASTAL, HARBOUR AND RIVER CONSULTANTS