# Efficient numerical algorithms for virtual design in nanoplasmonics

**A DISSERTATION**
**SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL**
**OF THE UNIVERSITY OF MINNESOTA**
**BY**

**Alexandra Ortan**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS**
**FOR THE DEGREE OF**
**Doctor of Philosophy**

**Fadil Santosa**

**March, 2017**

# Acknowledgements

I must first and foremost thank my adviser, Fadil Santosa, whose help and support were essential in the completion of this thesis. I am also eternally grateful for the guidance of Fernando Reitich, who first introduced me to the fascinating worlds of integral equations and nanoplasmonics. Last but not least, I thank the Natural Sciences and Engineering Research Council of Canada and the *Fonds de recherche du Québec* for their generous financial support during my graduate studies.

## Abstract

Nanomaterials have given rise to many devices, from high-density data storage to optical bio-sensors capable of detecting specific biochemicals. The design of new nanodevices relies increasingly on numerical simulations, driving a need for efficient numerical methods. In this work, integral equations are used to efficiently solve the electromagnetic transmission problem at the interface of a dielectric and a periodic metal nanostructure. Derivative-free trust-region methods are then used to optimize the geometry of the nanostructure.

# Contents

**Acknowledgements** **i**

**Abstract** **ii**

**List of Figures** **vi**

**1 Introduction** **1**

**2 Preliminaries** **5**

2.1 Maxwell's equations . . . . . 5

2.2 The optical characteristics of metals . . . . . 9

**3 A review of numerical methods in nanoplasmonics** **12**

3.1 Finite-difference methods . . . . . 12

3.2 Finite Element methods . . . . . 13

3.3 Integral equation methods . . . . . 14

**4 Integral equations solver: formulation and numerical implementation** **18**

4.1 Integral formulation . . . . . 19

4.2 Cylindrical domains . . . . . 22

4.3 Kernel decompositions . . . . . 25

4.4 Numerical solution . . . . . 26

4.5 Numerical results . . . . . 27

**5 Shape derivatives** **30**

5.1 Differential formulation . . . . . 30

5.2 Integral formulation . . . . . 34
5.3 Decomposition of the weakly singular kernels $K_5$, $K_6$ and $K_7$ . . . . . 37
5.3.1 Numerical evaluation of weakly singular integrals $I_5$, $I_6$, $I_7$, $I_9$ . . 42
5.4 Decomposition of the hyper-singular kernel $K_8$ . . . . . 43
5.5 Numerical evaluation of hypersingular singular integral $I_8$ . . . . . 44
5.6 Numerical integration with smooth kernel $K_{82}$ . . . . . 46

**6 Derivative-Free Optimization** **55**
6.1 Derivative-Free Trust Region Algorithm . . . . . 56
6.2 Implementation . . . . . 61
6.2.1 Choice of parameters . . . . . 61
6.2.2 Choice of model . . . . . 62
6.3 Validation . . . . . 62

**7 Optimal Design of Nanoplasmonic Surfaces** **67**
7.1 Objective functions . . . . . 67
7.1.1 Maximum point-wise surface field . . . . . 68
7.1.2 Minimal reflectivity . . . . . 68
7.2 Parameter search space . . . . . 69
7.3 Implementation . . . . . 70
7.4 Results . . . . . 71
7.4.1 Minimal reflectivity under normal incidence . . . . . 72
7.4.2 Minimal reflectivity under oblique incidence . . . . . 73
7.5 Discussion . . . . . 74

**8 Conclusion** **76**

**Bibliography** **79**

**Appendix A. Appendix** **86**
A.1 Hankel functions . . . . . 86
A.2 Smoothness of $sgn(t-\tau)\frac{dr(t,\tau)}{dt}$ . . . . . 88
A.3 Smoothness of $M(t,\tau)$ term in $K_{8,2}(t,\tau)$ kernel . . . . . 88
A.4 Evaluating $I(c)$ . . . . . 89

A.5 Guide to Variable Names . . . . . 96

# List of Figures


1.1 Glass containing gold or silver nanoparticles has different optical properties depending on the incidence of light. . . . . 2

1.2 A metal nanosphere subjected to an electric field will see its electrons displaced with respect to the lattice ions. . . . . 3

2.1 An infinitely periodic metal grating is illuminated by an incident field, giving rise to a scattered field (shown) and a transmitted field (not shown). 5

2.2 The Drude and the Drude-Lorentz models fitted to literature values of the dielectric data for gold. Figure and data from [1]. . . . . 10

2.3 The Drude-Lorentz model along with an interpolation of the measured values (from Palik's handbook [2]) of the dielectric constant for silver as a function of wavelength in microns. . . . . 11

4.1 A 2D periodic metal grating is illuminated by an incident field. . . . . 23

4.2 The total field (left) and the normal derivative (right) of the total field on the surface of the grating. Here $N = 32$. . . . . 28

4.3 The error in the total field (left) and its normal derivative (right) as a function of the number of points. The error is shown on a logarithmic scale. 28

4.4 The field above the surface of the grating. From left to right, the norm of the field, the real part and the imaginary part. . . . . 29

5.1 The Chebyshev interpolation of kernels $K_{5,1}$ and $K_{5,2}$ with 32 terms is accurate to 11 digits. Note the rapid decay of the Chebyshev coefficients for each kernel. . . . . 39

5.2 The Chebyshev interpolation of kernels $K_{6,1}$ and $K_{6,2}$ with 32 terms is accurate to 11 digits. Note the rapid decay of the Chebyshev coefficients for each kernel. . . . . 41

5.3 The Chebyshev interpolation of kernels $K_{7,1}$ and $K_{7,2}$ with 32 terms is accurate to 8 digits. Note the rapid decay of the Chebyshev coefficients for each kernel. . . . . 42
5.4 The Chebyshev interpolation of kernel $K_{8,2}$ with 16 and 32 terms is accurate only to 2 digits. The interpolations from the Chebfun package are identical to machine precision for 16 terms, but starts to differ from the manual interpolation with 32 terms. . . . . 47
5.5 With 64 terms, the Chebyshev interpolation of kernel $K_{8,2}$ is accurate only to 3 digits. The Chebfun overfits badly. With 128 terms, the accuracy is still around 3 digits. . . . . 50
5.6 Convergence error for integral $I_{8,2}$ using Floquet approximation of $\xi_1'$ and Chebyshev interpolation of kernel $K_{8,2}$ (right), versus trapezoidal rule directly on integral $W_{8,2}^n$ (left). . . . . 50
5.7 Convergence error for integral $I_{8,2}$ using Floquet approximation of $\xi_1'$ and Matlab's built-in adaptive quadrature quad (right), versus quadgk (left). 51
5.8 Convergence error for integral $I_{8,2}$ using the trapezoidal rule. . . . . 51
5.9 Convergence error for integral $I_{8,2}$ using a Chebyshev interpolation of the kernel $K_{8,2}(t,\tau)\xi_1'(\tau)$ (left), along with the interpolation error distribution on the interval $[-1,1]$ (right). . . . . 52
5.10 Convergence error for integral $I_{8,2}$ using a split Chebyshev interpolation of the kernel $K_{8,2}(t,\tau)\xi_1'(\tau)$ (left), along with the interpolation error distribution on both integration intervals (right). . . . . 52
5.11 Convergence error for integral $I_{8,2}$ using Matlab's built-in adaptive quadratures quad (left) and quadgk (right). . . . . 53
5.12 Numerical noise in the evaluation of the smooth kernel $K_{8,2}$. . . . . 53
5.13 Convergence error for integral $I_{8,2}$ using a split Chebyshev interpolation of the kernel $K_{8,2}(t,\tau)\xi_1'(\tau)$ (left), along with the interpolation error distribution on both integration intervals (right). . . . . 54
5.14 Convergence error for integral $I_{8,2}$ using Matlab's built-in adaptive quadratures quad (left) and quadgk (right). . . . . 54

6.1 Convergence of the DFO algorithm on a quadratic test function in nine dimensions. The plot on the left shows the iterates on a two-dimensional cross-section of the contour plot of the function, while the plot on the right shows that machine precision is reached within a single iteration, requiring a total of 101 function calls. . . . . 63
6.2 Convergence of the DFO algorithm on a quadratic test function in two dimensions. The plot on the left shows the iterates on a two-dimensional cross-section of the contour plot of the function, while the plot on the right shows that machine precision is reached within two iteration, requiring a total of 13 function calls. . . . . 64
6.3 Convergence of the DFO algorithm on the Golstein-Price test function in two dimensions. The plot on the left shows the iterates on a two-dimensional cross-section of the contour plot of the function, while the plot on the right shows that machine precision is reached within two iteration, requiring a total of 127 function calls. . . . . 65
6.4 Convergence of the DFO algorithm on the Beale test function in two dimensions. The plot on the left shows the iterates on the contour plot of the function, which is mostly flat except for a sharp dip at the origin. while the plot on the right shows that machine precision is reached with 279 function calls. . . . . 66
6.5 Convergence of the DFO algorithm on the Rosenbrock test function in two dimensions. The plot on the left shows the iterates on a contour plot of the function, while the plot on the right shows that machine precision is reached with 413 function calls. . . . . 66
7.1 The DFO algorithm uses less than 600 evaluations of the objective function to reduce the reflectivity to machine precision 0. The corresponding grating profile, on the right, has a total height of around 40nm. . . . . 72
7.2 The field in a 1300nm band around the grating confirms the conclusion of the DFO algorithm: most of the energy is concentrated very close to the surface, where the field is highly enhanced. . . . . 73

7.3 The DFO algorithm uses 1000 evaluations of the objective function to reduce the reflectivity to $10^{-2.5}$. The corresponding grating profile, on the right, has a total height of 100nm. . . . . 74
7.4 The field in a 1300nm band around the grating confirms the conclusion of the DFO algorithm: most of the energy is concentrated very close to the surface, where the field is highly enhanced. . . . . 75
A.1 Contour path used to compute complex integral $I(c)$. . . . . 91
A.2 Convergence of the numerical approximation to $I(0)$. . . . . 93
A.3 Convergence of the numerical approximation to $I(-1/2)$, using respectively 16 and 100 digits of accuracy. . . . . 94
A.4 Convergence of the numerical approximation to $I(1/2)$. . . . . 95

# Chapter 1

# Introduction

The field of nanoplasmonics studies the confinement of electromagnetic fields on sub-wavelength scales at the interface of metals and dielectrics, which is possible due to interactions between the electromagnetic radiation and conducting electrons within the metal surface. Though these phenomena had first been documented almost a century ago, nanoplasmonics has generated much interest in recent years due to an increasing number of applications such as bio-sensing, nanoscopy and optical data storage.

The interesting optical properties generated by the interactions between visible light and the certain metal powders, such as gold or silver, have been known at least since Roman times. They are responsible among other things for the visual effects exhibited by the Lycurgus cup, a 4th century Roman glass cup, and the stained glass windows in the Sainte Chapelle in Paris [3]. In the case of the Lycurgus cup (see figure 1.1a), gold-silver alloy nanocrystals embedded within the glass cause it to appear green when light is reflected from it, and red when light is transmitted through. Similar nanoparticles give the stained glass windows of the Sainte Chapelle a red glow whose intensity depends on the incident and viewing angles (see figure 1.1b).

However, it wasn't until 1908 that Mie gave a mathematical description of light scattering from spherical particles of sizes comparable to the wavelength [4], describing an effect that will come to be known as localized surface plasmons in the context of nanoplasmonics. In 1899, Sommerfeld had described surface waves (waves propagating at the surface of metals) mathematically, and in 1902 Wood observed anomalous drops in the intensity of light reflected by a metallic grating [5]. But theory and observation would

not be linked until 1941, by Fano [6]. Further experimental validation came in 1968, when Kretschmann and Raether used prism coupling to excite surface waves with visible light [7]. Thus far, surface waves had only been observed on continuous surfaces. This changed in 1998, when Ebessen et al. [8] discovered Extraordinary Optical Transmission (EOT) through sub-wavelength hole arrays for specific frequencies of incident light. The link with surface waves was first established by Popov et al. [9], who used (the Fourier modal method in) numerical simulations to successfully reproduce the experimental results of Ebessen, and has since been validated by many others [10].

All of the phenomena mentioned above are based entirely on classical electromagnetics, and thus can be mathematically described by Maxwell's equations. Physically, they can be described as the coupling of an electromagnetic field with collective charge oscillations at the surface of a metal. If a metal nanostructure is smaller than the metal's skin depth, the field penetrates it throughout and causes the free electrons to oscillate around the lattice ions, as shown in figure 1.2. The charge difference created by the displaced electrons means there is also a restoring force, resulting in an electron oscillator, whose quantum is called a surface plasmon (SP) [3]. When coupled to a photon, these oscillations propagate as surface waves, known as surface plasmon polaritons (SPP). SPPs are effectively surface charge density oscillations that decay exponentially away

(a) The Lycurgus cup in reflected light on the left, and in transmitted light on the right.

(b) The light transmitted through the Sainte Chapelle stained glass windows has different intensities depending on the angle of incidence.

Figure 1.1: Glass containing gold or silver nanoparticles has different optical properties depending on the incidence of light.

from the surface [7]. When the charge oscillations are not propagating, but instead are localized in space (as in the case of metallic nanospheres), they are called localized surface plasmons (LSP). For both SPPs and LSPs, the resonant frequencies depend highly on the refractive index of the surrounding material, a very important property for a number of applications.

Indeed, the relevance of nanoplasmonics is closely connected with the wealth of actual and potential applications that they enable. For instance, bio-sensors such as the home pregnancy test use a latex substrate covered with a layer of gold nanoparticles linked to antibodies which target hCG, a human hormone present during pregnancy. When the hCG binds to the antibodies, it effectively changes the refractive index of the dielectric medium above the gold nanoparticle array, thus shifting its resonant frequency and causing a change of the scattered color under natural incident light. The method can be extended to detect other medical conditions, such as HIV, prostate cancer and heart attacks [3].

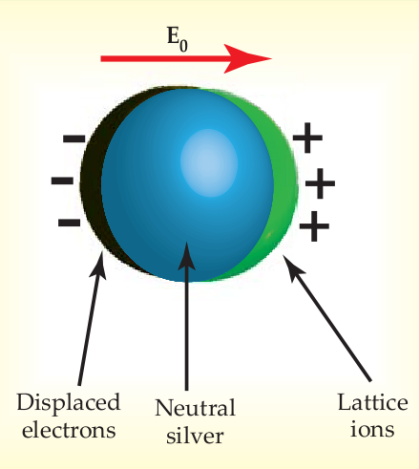



Figure 1.2: A metal nanosphere subjected to an electric field will see its electrons displaced with respect to the lattice ions.

The transmission properties of nanohole arrays can also be exploited for high-density optical data storage. This application takes advantage of nanohole and nanorod arrangements with frequency-dependent or polarization-dependent transmission. Thus, illuminating the same area with blue or red light, or changing the polarization would show different scattered images, leading to five-dimensional optical data storage [11].

Most applications use either gold or silver for the metallic nanostructures, due to their low losses in the optical domain. Though silver has the lowest loss and is the least onerous, corrosion makes it an ill-suited choice in certain environments. Oh et al. [12] offer a more in-depth review of material choices as well as the fabrication techniques for metallic nanostructures.

While some of the applications have already been commercialized, many are still

being developed, and this development relies increasingly on numerical simulations. Of particular interest are numerical algorithms capable of optimizing relevant properties of the nanostructure designed. The methods used so far, an overview of which will be given in chapter 3, have been successful, but can take very long to simulate a given setup. It is the goal of this thesis to present a integral equation-based numerical algorithm fast enough to be used for optimization. A complete optimization package is to be developed in the future.

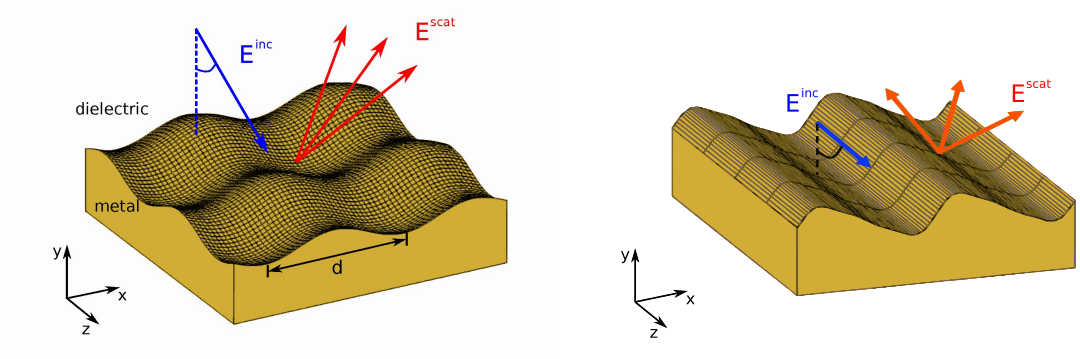
The thesis is organized as follows: first some preliminary concepts and equations describing the physics of surface plasmons are discussed in chapter 2. Chapter 3 reviews the various numerical methods in use in the field of nanoplasmonics. The proposed numerical algorithm is introduced in chapter 4, which also offers some of the details of the numerical implementation and the results of a few numerical experiments, validating the proposed numerical algorithm. Chapter 5 explores shape derivatives and efficient ways to evaluate them, with a view to use in optimization. Chapter 6 introduces a derivative-free optimization algorithm as an alternative to the shape derivative based ones, and discusses practical implementation challenges and solutions. Finally, chapter 7 takes the methods introduced previously and combines them into an end-to-end virtual design algorithm, and shows that this algorithm leads to good design outcomes with unprecedented speed.

# Chapter 2

# Preliminaries

## 2.1 Maxwell's equations

The setup that will be studied throughout this thesis consists of an infinitely periodic metal surface whose period $d$ is on the nanometer scale. The metal is assumed to extend infinitely below this surface, while a dielectric material extends infinitely above the surface, as shown in figure 2.1. An incident field illuminates the metal surface from above, giving rise to scattered and transmitted fields. The goal here will be to find the scattered and transmitted fields given information about the surface and the incident fields.




(a) Three dimensional grating.

(b) Two dimensional grating.

Figure 2.1: An infinitely periodic metal grating is illuminated by an incident field, giving rise to a scattered field (shown) and a transmitted field (not shown).

As mentioned in the introduction, the behavior of nanoplasmonic effects falls entirely in the realm of classical electromagnetic theory, and is thus governed by Maxwell's equations for the electric and magnetic fields $\mathbf{E}$ and $\mathbf{H}$:

$$\begin{aligned}
\nabla \cdot \mathbf{E} &= \frac{1}{\epsilon}\rho_f, \\
\nabla \times \mathbf{E} &= -\mu \frac{\partial \mathbf{H}}{\partial t}, \\
\nabla \cdot \mathbf{H} &= 0, \\
\nabla \times \mathbf{H} &= \mathbf{J}_f + \epsilon \frac{\partial \mathbf{E}}{\partial t}.
\end{aligned}$$

Here the equations are written for linear materials, where $\epsilon$ and $\mu$ are the real permittivity and permeability of the material respectively, and $\rho_f$ and $\mathbf{J}_f$ are the free charge and current densities respectively [13].

In the present setup, there are two domains of interest: the region below the grating, $D$ (the metal scatterer), and the region above the grating, $D^C$ (the dielectric). The corresponding quantities shall then be indicated by the superscript $i$ on $D$ and $e$ on $D^C$, for interior and exterior with respect to $D$. Both the metal and the dielectric materials are assumed to have the permeability $\mu_0$ of free space. Since the upper region is an insulator, it carries neither free charges nor free currents. On the other hand, the lower region has non-zero conductivity $\sigma$ and therefore will display a free current proportional to the electric field $\mathbf{J}_f = \sigma \mathbf{E}$. Any free charge inside the conductor will quickly dissipate and may be assumed to be 0. To summarize, the fields interior and exterior to $D$ obey

$$\begin{aligned}
\nabla \cdot \mathbf{E}^e &= 0, & \nabla \cdot \mathbf{E}^i &= 0, \\
\nabla \times \mathbf{E}^e &= -\mu_0 \frac{\partial \mathbf{H}^e}{\partial t}, & \nabla \times \mathbf{E}^i &= -\mu_0 \frac{\partial \mathbf{H}^i}{\partial t}, \\
\nabla \cdot \mathbf{H}^e &= 0, & \nabla \cdot \mathbf{H}^i &= 0, \\
\nabla \times \mathbf{H}^e &= \epsilon^e \frac{\partial \mathbf{E}^e}{\partial t}, & \nabla \times \mathbf{H}^i &= \sigma \mathbf{E}^i + \epsilon^i \frac{\partial \mathbf{E}^i}{\partial t}.
\end{aligned}$$

Since it depends on time, the system above is said to be in the time domain. In the context of nanoplasmonics however, the relevant information is encoded in the frequency response of the system. As such, it is convenient recast the equations in frequency, or Fourier domain by assuming that each field has a $e^{-i\omega t}$ time dependence. Thus the fields

interior and exterior to $D$ satisfy the time harmonic Maxwell's equations:

$$\nabla \cdot \mathbf{E}^e = 0, \qquad \nabla \cdot \mathbf{E}^i = 0, \tag{2.3a}$$

$$\nabla \times \mathbf{E}^e = i\omega\mu_0 \mathbf{H}^e, \qquad \nabla \times \mathbf{E}^i = i\omega\mu_0 \mathbf{H}^i, \tag{2.3b}$$

$$\nabla \cdot \mathbf{H}^e = 0, \qquad \nabla \cdot \mathbf{H}^i = 0, \tag{2.3c}$$

$$\nabla \times \mathbf{H}^e = -i\omega\epsilon^e \mathbf{E}^e, \qquad \nabla \times \mathbf{H}^i = -i\omega\epsilon^{metal} \mathbf{E}^i, \tag{2.3d}$$

where $\epsilon^{metal} = \epsilon^i + i\frac{\sigma}{\omega}$ is the complex permittivity inside $D$.

To completely specify the problem, note that for each of the fields, the components parallel to the surface must be continuous, while the components normal to the surface are continuous for $\mathbf{H}$ and jump by a multiple of $\epsilon^i - \epsilon^e$ for $\mathbf{E}$. This gives rise to the following boundary conditions:

$$\epsilon^e \mathbf{n} \cdot \mathbf{E}^e = \epsilon^i \mathbf{n} \cdot \mathbf{E}^i, \qquad \mathbf{n} \times \mathbf{E}^e = \mathbf{n} \times \mathbf{E}^i, \tag{2.4a}$$

$$\mathbf{n} \cdot \mathbf{H}^e = \mathbf{n} \cdot \mathbf{H}^i, \qquad \mathbf{n} \times \mathbf{H}^e = \mathbf{n} \times \mathbf{H}^i, \tag{2.4b}$$

where $\mathbf{n}$ is the unit normal to the boundary surface.

Equations 2.3 can be reduced to two Helmholtz equations for the $\mathbf{E}$ and $\mathbf{H}$ fields:

$$\Delta \mathbf{E} + k^2 \mathbf{E} = 0,$$
$$\Delta \mathbf{H} + k^2 \mathbf{H} = 0,$$

where $k = k_e = \omega\sqrt{\mu_0 \epsilon^e}$ outside of $D$ and $k = k_i = \omega\sqrt{\mu_0 \epsilon^{metal}}$ in $D$. Note that this does not imply that $\mathbf{E}$ and $\mathbf{H}$ are decoupled, since the boundary conditions are not. This form will be more convenient to work with since the Green's function of the Helmholtz equation is known, allowing for the integral formulation to be used, as will be discussed shortly.

An interesting observation at this point is the decoupling that takes place in the special case of a 2D grating, shown in figure 2.1b. If one of the spatial dimensions (say the $\hat{z}$ direction, see figure 2.1b) in the problem is infinitely long, then the equations may be reduced to two dimensions by assuming all quantities are constant in the $\hat{z}$ direction. In this case, the equations decouple in two systems: one for $H_x$, $H_y$ and $E_z$ (the TE

mode of polarization) and one for $E_x$, $E_y$ and $H_z$ (the TM mode of polarization):

$$\begin{aligned}
\frac{\partial H_x}{\partial x} + \frac{\partial H_y}{\partial y} &= 0, & \frac{\partial E_x}{\partial x} + \frac{\partial E_y}{\partial y} &= 0, \\
\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} &= -i\omega\epsilon E_z, & \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} &= i\omega\mu_0 H_z, \\
i\omega\mu_0(-H_y, H_x) &= \left(\frac{\partial E_z}{\partial x}, \frac{\partial E_z}{\partial y}\right), & -i\omega\epsilon(-E_y, E_x) &= \left(\frac{\partial H_z}{\partial x}, \frac{\partial H_z}{\partial y}\right),
\end{aligned}$$

where $\epsilon = \epsilon^e$ outside $D$ and $\epsilon = \epsilon^{metal}$ inside $D$. Note that in this system the $x$ and $y$ components of each field are easily obtained once the $z$ components are known. Substituting the expressions for $E_x$, $E_y$, $H_x$ and $H_y$ from the last two equations into the two equations above yields two Helmholtz equations for $E_z$ and $H_z$:

$$\begin{aligned}
\Delta E_z + k^2 E_z &= 0, \\
\Delta H_z + k^2 H_z &= 0,
\end{aligned}$$

Boundary conditions for $E_z$ and $H_z$ are obtained by observing that if the surface extends infinitely in the $\hat{\mathbf{z}}$ direction, then $\mathbf{n} \cdot \hat{\mathbf{z}} = 0$. Thus the boundary conditions (2.4a) and (2.4b) on the parallel components of $\mathbf{E}$ and $\mathbf{H}$ imply:

$$\begin{aligned}
E_z^e &= E_z^i, \\
H_z^e &= H_z^i.
\end{aligned}$$

Boundary conditions for $\frac{\partial E_z}{\partial \mathbf{n}}$ and $\frac{\partial H_z}{\partial \mathbf{n}}$ are obtained by taking the dot product of $\hat{\mathbf{z}}$ with boundary conditions (2.4b) and (2.4a) respectively, then using equations (2.3b) and (2.3d) respectively:

$$\begin{aligned}
\frac{\partial E_z^e}{\partial \mathbf{n}} &= \frac{\partial E_z^i}{\partial \mathbf{n}} \\
\frac{\partial H_z^e}{\partial \mathbf{n}} &= \frac{\epsilon^e}{\epsilon^{metal}} \frac{\partial H_z^i}{\partial \mathbf{n}}.
\end{aligned}$$

In a scattering problem, the incident (known) and the scattered (unknown) parts of the exterior fields can be distinguished: $E_z^s + E_z^{inc} = E_z^e$ and $H_z^s + H_z^{inc} = H_z^e$. The field inside the metal is referred to as the transmitted field and will be denoted by $E_z^t$, $H_z^t$.

Thus $E_z$ (TE mode) and $H_z$ (TM mode) both satisfy the following system

$$\Delta u^t + k_i^2 u^t = 0 \qquad \text{on } D, \tag{2.6a}$$

$$\Delta u^s + k_e^2 u^s = 0 \qquad \text{on } D^C, \tag{2.6b}$$

$$\Delta u^{inc} + k_e^2 u^{inc} = 0 \qquad \text{on } D^C, \tag{2.6c}$$

$$u^s + u^{inc} = u^t \qquad \text{on } \partial D, \tag{2.6d}$$

$$\nu \frac{\partial}{\partial n}[u^s + u^{inc}] = \frac{\partial u^t}{\partial n} \qquad \text{on } \partial D, \tag{2.6e}$$

where $\nu = 1$ for the TE mode, and $\nu = \frac{\epsilon^{metal}}{\epsilon^e} = \frac{k_i^2}{k_e^2}$ for the TM mode.

To ensure the uniqueness of a solution to this system, another "boundary" condition infinitely away from the surface is needed on $u^s$. Physically, this condition must guarantee that the scattered field is propagating away from the surface. Mathematically, it can be enforced by requiring that above a certain point, $u^s$ be a linear superposition of upward propagating plane waves [14, 15]. The uniqueness result for the two-dimensional problem where $\Im(\nu^2) > 0$ has been proved in [16]. This condition is sufficient here, as it holds in plasmonics applications, as shall be seen in the next section.

## 2.2 The optical characteristics of metals

As mentioned before, the dielectric constant of a metal is in general complex: $\epsilon^{metal} = \epsilon^i + i\frac{\sigma}{\omega}$. For metals, the real part of $\epsilon^{metal}$ is negative while the imaginary part is positive and determines the amount of absorption. Low values of $Im[\epsilon^{metal}]$ translate in low losses, a key aspect of the material choice in nanoplasmonic applications. Silver for example has the lowest losses in the visible and infrared, with gold coming in second for wavelengths above 600nm [12].

Since the response of metals depends on the frequency $\omega$, a model describing this dependence is needed for simulation purposes. The most widely used is the Drude model, which models the free electrons in a metal as a plasma, and gives the dielectric function of the frequency as [7]:

$$\epsilon(\omega) = 1 - \frac{\omega_p^2}{\omega^2 + i\gamma\omega},$$

where $\omega_p$ is the plasma frequency of the free electron gas and $\gamma$ is the characteristic collision frequency. Typically, these quantities are obtained as parameter fits of experimental values. For noble metals, this model is only valid up to visible frequencies, as can be seen from figure 2.2. Beyond this point, interband transitions in the atoms becomes significant and the Lorentz extension of the model is used instead. The Drude-Lorentz model accounts for the interband transitions by adding for each one of them a Lorentz-oscillator term of the form $\frac{A_i}{\omega_i^2 - \omega^2 - i\gamma_i\omega}$, where $A_i$, $\omega_i$ and $\gamma_i$ are properties of the oscillator are obtained by a fit of the data [7].

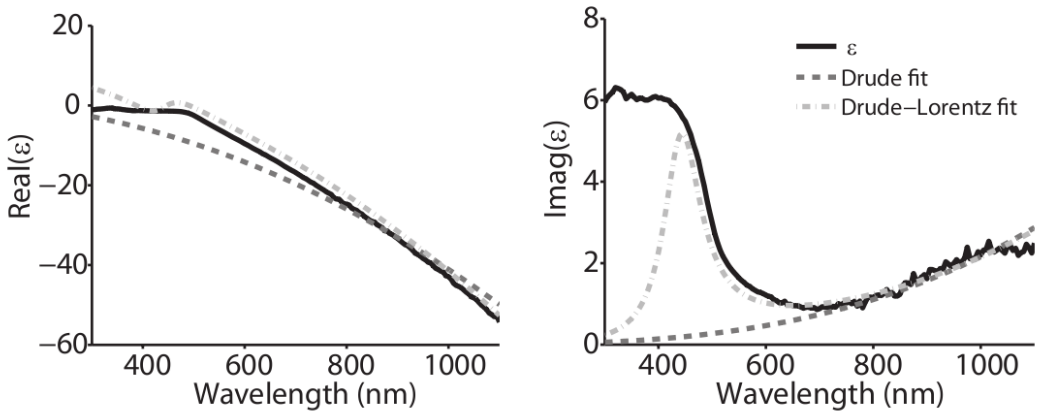



Figure 2.2: The Drude and the Drude-Lorentz models fitted to literature values of the dielectric data for gold. Figure and data from [1].

A more straight-forward way of modeling the dielectric constant however is simply to interpolate the experimental values. The advantage of an interpolation is that it is not limited to a particular range of frequencies. Figure 2.3 shows that the interpolation is in fact a better fit for the experimental values than the Drude-Lorentz model.

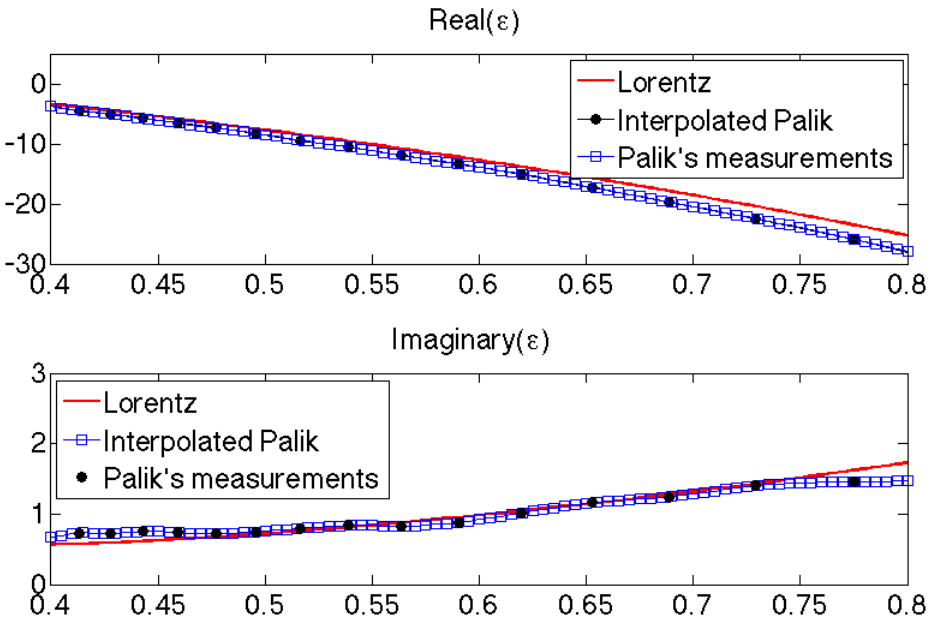



Figure 2.3: The Drude-Lorentz model along with an interpolation of the measured values (from Palik's handbook [2]) of the dielectric constant for silver as a function of wavelength in microns.

# Chapter 3

# A review of numerical methods in nanoplasmonics

Direct design of nanoplasmonic structures is not practical due to high fabrication costs, particularly in terms of time. Thus successful design depends heavily on the use of appropriate numerical methods for the simulation of electromagnetic fields in the vicinity of the structure. A number of methods have been used so far in the field, with finite difference methods being by far the methods of choice [12, 17, 18, 19], followed by finite element methods [20, 21, 22, 23], and recently also by integral methods [24, 25, 26, 27].

This chapter offers a brief review of these methods in the context of nanoplasmonics.

## 3.1 Finite-difference methods

Finite-difference methods use a direct discretization of Maxwell's differential equations using difference quotients. The most popular of such methods, the Finite Difference Time Domain Method (FDTD), was introduced in 1966 by Yee [28] and popularized by A. Taflove a few years later[29]. It uses the Yee cell as a building block for a staggered space-time grid for the electric and magnetic field variables. The time derivatives are approximated by a second-order centered difference, which makes the method explicit in time. The convergence is second order in both space and time.

While the method has an enduring popularity due to its ease of implementation and parallelization [30], it also has important drawbacks. One of them is that Cartesian

grids must be used, which gives rise to a stair-casing effect around curved interfaces. To maintain a desired accuracy, a fine mesh must then be used, and thus a small time-step (in order to satisfy the Courant-Friedrichs-Levy stability condition: $\Delta t \leq \Delta x/c$), leading to a high computational cost. The use of conformal meshes has been observed to decrease computational effort by a factor of 4, for a fixed accuracy ([31], [32]), and has further been accelerated using parallel processing [33]. However this does not fully avoid the problem, since cells near the interface must still be smaller, and thus still reduce the global time-step. Another way to avoid the stair-casing effect is to use a hybrid finite element/finite difference mesh, where finite elements are used around the curved boundary, and interfaced with the finite difference grid away from the boundary. Numerical experiments have shown this approach to yield better results than the original FDTD[34] in some cases. No theoretical results have been proven at this time though.

Clearly an important bottle-neck of the FDTD method is the CFL stability condition. To avoid such a strict condition on the time steps, an alternating-direction implicit (ADI) technique for solving PDE's was introduced in [35] and [36], yielding an unconditionally stable version of FDTD. However, this only reduces the computational complexity of the problems if the time-period is larger than the length scale of the object modeled, since the time step still needs to resolve the period of the signal.

Further, since FDTD algorithms discretize Maxwell's equations directly, they need to compute the fields in the infinite domain outside the scatterer. As this is not feasible numerically, the domain must be truncated. Since the new boundary thus introduced is artificial, some effort must be made to render it effectively invisible to incoming or outgoing waves. This is achieved by using absorbing boundary conditions [37, 38, 39].

## 3.2 Finite Element methods

Finite element methods (FEM) are based on the weak formulation of Maxwell's equation. [40] They discretize space using an irregular mesh, typically using simplices as their building blocks. The unknowns and the test functions are expanded in terms of basis functions with support on only a few adjacent mesh elements. This leads to a sparse linear system, which can be solved in $\mathrm{O}(N)$ time, where $N$ is the number of unknowns.

Different methods use different function spaces for the basis functions, but piecewise

polynomial spaces are the most common. The function spaces can have additional continuity conditions, either for the values of the functions themselves, or other degrees of freedom across the boundary between elements. This approach is known as the continuous Galerkin method. Another approach, known as the discontinuous Galerkin method, does not enforce these continuity conditions directly, but instead uses penalty terms on the boundaries between elements. This seems to lead to more stable methods, and can still achieve high accuracy. [41]

Both the continuous [20, 21] and discontinuous approaches [22, 23] have been successfully used in the context of nanoplasmonics. Their popularity is due to the two major advantages they offer over the FDTD method. The first is that FEM can achieve higher orders of convergence than FDTD, and the order can be controlled by the degree of the polynomial spaces used. The second and perhaps most important is that FEM meshes can be adaptive, which means the number of unknowns can be vastly reduced, especially in the context of nanoplasmonics where the fields are confined very close to the nanostructures, and thus only a small region requires the use of a fine mesh [20]. Moreover, the irregular meshes can discretize complex domains much better than FDTD, avoid stair-casing effects.

However, as with the FDTD method, FEM needs to discretize the domain exterior to the scaterrer, leading to artificial boundaries which need to be treated with special care. Absorbing boundary conditions have been developed for FEM, both for the wave equation [42] and for Maxwell's equations [43]. However, the implementation of these conditions requires rectangular boundaries for optimal results [24], which may result in an unnecessarily large computation domain.

## 3.3 Integral equation methods

Though integral equations have traditionally been developed in the context of diffraction gratings, it is only in recent years that the nanoplasmonics community has started to use integral methods, both on their own [24, 25] or in conjunction with FE methods [27, 26].

Integral methods are based on an integral formulation of Maxwell's equations. Two categories can be distinguished: volumetric and surface integral equations, depending on whether the integrals are posed over the volume of the scatterer or only its surface.

In both cases, the key difficulty lies in the accurate and fast evaluation of one or more integrals of the form

$$\int_{\Omega} K(x,y)\phi(y)dy$$

where $\phi$ is the unknown of interest, $\Omega$ is a volume or a surface, as the case may be, and $K(x,y)$ is a singular kernel. One of the advantages of this approach is that the radiation conditions are encoded in the choice of the kernel, thus avoiding the issue of enforcing the radiation condition at the boundary of the computation domain. The other advantage is the speed and accuracy of these methods, as shall be discussed below.

Just as with any other method, these formulations lead to linear systems which are typically solved with the use of iterative methods. Thus the complexity is limited only by the complexity of the matrix-vector multiplication. This can be achieve in a number of ways, the most popular of which being divided in two classes: methods of moments (MoM) and Nyström methods. The MoM's involve approximating the solution by a finite number of basis function, and requiring the integral equation to be solved in the weak sense over the span of finitely many test functions. Depending on the basis and test functions used, this approach leads to the collocation method (deltas for test functions) or the Galerkin method (same basis and test functions) [44]. The Nyström methods on the other hand rely on a numerical quadrature of the integral, leading again to a linear system. Whenever applicable, Nyström methods are preferable, as they require the least computational effort [45].

In the case of volumetric integral methods, the unknown solves the so-called "Lippmann-Schwinger integral equation" over the volume of the scatterer [46]. Thus the computation domain is reduced while still allowing for inhomogeneities in scatterer. Despite the reduction of the computation domain however, an improved complexity is not guaranteed, since the evaluation of the integral gives rise to a full matrix of size $\mathrm{O}(n^3)$, where $n$ is the number of discretization points taken in each of the three spatial dimensions. Naively, one might expect the complexity of the integration to be $\mathrm{O}(n^6)$, which cannot compete with a FEM, where the same problem takes $\mathrm{O}(n^3)$ steps. However the cost of the integration can be reduced to $\mathrm{O}(n^3 \log n^3)$ [47], while achieving higher than second order of convergence of the solution.

On the other hand, if the scatterer is piecewise homogeneous, as is the case in the

nanoplasmonic applications of interest, a surface integral formulation can be used instead. This integral formulation can be derived from Green's theorem or from layer potentials [48], and can be expressed either as a first kind or a second kind integral equation. In either case, the computation domain is reduced to the surface of the scatterer, leading to a full matrix of size $O(n^2)$ in 3D, which a priori requires $O(n^4)$ steps to apply - not competitive with FEM matrices. As in the case of volumetric integrals however, this bound can be drastically improved by the use of acceleration methods such as fast multipole methods (FMM) [49, 50] and FFT-based methods [51, 47, 52, 53].

Before discussing the different acceleration methods, a note on the choice of surface integral formulation is necessary. Though some authors use the first kind integral equation [54], comparisons of the two [55] clearly show that second kind integral equations are preferable in terms of speed and accuracy. This is easily seen in the context of Fredholm theory: the solution to a first kind integral equation requires the inversion of a compact operator, whose eigenvalues approach 0, making numerical inversion more unstable, while the solution to a second kind integral equation requires the inversion of the identity plus a compact operator, whose eigenvalues approach 1.

The multi-level fast multipole method, introduced in [49] for integral equations for the 2D Laplace's equation, was able to accelerate the integral evaluation to $O(N)$ operations, for a total number $N$ of points. The idea behind these methods is to divide the domain into clusters and treat sources in a single cluster as a single source when viewed from far away. This is accomplished using multipole expansions of the Green's function, effectively reducing the full matrix representing the integral equation to a block-wise low-rank matrix [56]. Since it has been introduced, the method has been extended to 3D Helmholtz equation, where, together with carefully chosen preconditioners, it can achieve $O(N \log N)$ complexity [57]. The method has also been shown to accelerate the 3D Maxwell's equations for a 2-dimensional periodic structure [58], using spherical harmonics series expansion of the quasi-periodic Green's function. However, the applicability of the FMM is limited, since instabilities appear when the size of each cluster becomes much smaller than the wavelength of the incident field[59] .

A popular FFT-based method is the adaptive integral method (AiM), introduced in [51]. The method also treats near-field and far-field interactions differently, but instead of dividing the domain into clusters, it divides it using a regular grid. The components

of the interaction matrix are computed using a Galerkin discretization with locally supported basis functions, thus reducing the problem to only the interactions between the nodes of the grid. The advantage of this procedure is that the regular grid makes it possible to use a FFT to compute the interactions. This leads to an algorithm of complexity $\mathrm{O}(N \log N)$ and $\mathrm{O}(N^{3/2} \log N)$ in 2D and 3D respectively. The original AIM has been extended to the case of scattering from periodic arrays by using so-called characteristic basis functions [60].

The idea of using regular grids in order to take advantage of the power of FFT has also been exploited by [61, 47]. Instead of using equivalent point sources at the nodes of the grid, the authors propose to use equivalent surface sources on the faces of the grid in 3D. This approach reduces the computational complexity of the algorithm to $\mathrm{O}(N^{4/3} \log(N))$, but more interestingly it has exponential convergence, an important improvement over AiM.

# Chapter 4

# Integral equations solver: formulation and numerical implementation

Integral equations methods rely on recasting the differential equation description of the scattering problem (system (2.6a)) as a system of integral equations, and solving the latter. As mentioned before, the advantage of this approach is that it reduces the dimensionality of the problem, but a drawback is that the matrix that must be inverted is full. Further, the evaluation of the integral kernels themselves offers some challenges. These issues, along with the approaches used to overcome them, will be addressed in this chapter.

## 4.1 Integral formulation

In this section, a system of integral equations for the total exterior field $u$ and its normal derivative $\frac{\partial u}{\partial \mathbf{n}}$ on the surface $\partial D$ is derived from the equivalent differential system (2.6a):

$$\Delta u^t + k_i^2 u^t = 0 \qquad \text{on } D, \tag{2.6a}$$

$$\Delta u + k_e^2 u = 0 \qquad \text{on } D^C, \tag{2.6b}$$

$$u = u^t \qquad \text{on } \partial D, \tag{2.6d}$$

$$\nu \frac{\partial u}{\partial n} = \frac{\partial u^t}{\partial n} \qquad \text{on } \partial D, \tag{2.6e}$$

where $\nu = \frac{k_i^2}{k_e^2}$ and the total exterior field $u = u^s + u^{inc}$ is the sum of the scattered field $u^s$ and an incident field $u^{inc}$. Recall that $D$ denotes the interior domain with wave number $k_i$, while $D^C$ denotes the exterior domain with wave number $k_e$, and that the surface normal $\vec{n}$ is taken to point away from the domain $D$.

For a fixed point $x \in \mathbb{R}^n$ the free space Green's functions

$$\Phi^i(x,y) = \frac{i}{4} H_0^{(1)}(k_i|x-y|) \quad \text{in 2D}, \tag{4.1a}$$

$$\Phi^i(x,y) = \frac{e^{ik_i|x-y|}}{4\pi|x-y|} \quad \text{in 3D}, \tag{4.1b}$$

are at least twice differentiable in $y$ and satisfy the interior Helmholtz equation 2.6a in $\mathbb{R}^n \setminus \{x\}$. Similarly, the free space Green's functions

$$\Phi^e(x,y) = \frac{i}{4} H_0^{(1)}(k_e|x-y|) \quad \text{in 2D}, \tag{4.2a}$$

$$\Phi^e(x,y) = \frac{e^{ik_e|x-y|}}{4\pi|x-y|} \quad \text{in 3D} \tag{4.2b}$$

are at least twice differentiable in $y$ and satisfy the exterior Helmholtz equation 2.6b in $\mathbb{R}^n \setminus \{x\}$. Hence they can be used along with $u^t$, $u^s$, $u^{inc}$ in Green's second theorem in a region excluding a small ball around the fixed point $\{x\}$. Taking limits as the ball

shrinks in around $\{x\}$, the following Helmholtz representations are obtained [46]:

$$u^t(x) = \int_{\partial D} \left\{ \Phi^i \frac{\partial u^t(y)}{\partial \mathbf{n}_y} - u^t(y) \frac{\partial \Phi^i}{\partial \mathbf{n}_y} \right\} ds(y) \qquad x \in D, \tag{4.3a}$$

$$u^s(x) = \int_{\partial D} \left\{ u^s(y) \frac{\partial \Phi^e}{\partial \mathbf{n}_y} - \Phi^e \frac{\partial u^s(y)}{\partial \mathbf{n}_y} \right\} ds(y) \qquad x \in D^C, \tag{4.3b}$$

$$0 = \int_{\partial D} \left\{ u^{inc}(y) \frac{\partial \Phi^e}{\partial \mathbf{n}_y} - \Phi^e \frac{\partial u^{inc}(y)}{\partial \mathbf{n}_y} \right\} ds(y) \qquad x \in D^C, \tag{4.3c}$$

where $\mathbf{n}$ is the unit normal to $\partial D$ directed into the exterior of $D$ (i.e. "upward").

Observe that the previous set of equations explicitly give the interior and exterior fields in their respective domains, as long as $u(y) = u^s(y) + u^{inc}(y)$ and $\frac{\partial u}{\partial \mathbf{n}_y}(y)$ are known for $y \in \partial D$. Indeed, using the boundary conditions (2.6d) and (2.6e), and subtracting the third from the second,

$$u^t(x) = \int_{\partial D} \left\{ \Phi^i \nu \frac{\partial u(y)}{\partial \mathbf{n}_y} - u(y) \frac{\partial \Phi^i}{\partial \mathbf{n}_y} \right\} ds(y) \qquad x \in D, \tag{4.4a}$$

$$u^s(x) = \int_{\partial D} \left\{ u(y) \frac{\partial \Phi^e}{\partial \mathbf{n}_y} - \Phi^e \frac{\partial u(y)}{\partial \mathbf{n}_y} \right\} ds(y) \qquad x \in D^C. \tag{4.4b}$$

Thus it is sufficient to solve for $u(y)$ and $\frac{\partial u}{\partial \mathbf{n}(y)}(y)$ on the boundary of $D$ from the system of equations. To do so consider the equations (4.4) as $x \to \partial D$. Given that the single layer potential is continuous across the surface, and using the jump condition for the double layer potential ($+\frac{1}{2}u$ when taking limits from the outside and $-\frac{1}{2}u$ when taking limits from the inside [46]), the following hold:

$$\frac{1}{2}u(x) = \int_{\partial D} \left\{ \nu \Phi^i \frac{\partial u(y)}{\partial \mathbf{n}_y} - u(y) \frac{\partial \Phi^i}{\partial \mathbf{n}_y} \right\} ds(y) \qquad x \in \partial D, \tag{4.5a}$$

$$\frac{1}{2}u(x) - u^{inc}(x) = \int_{\partial D} \left\{ u(y) \frac{\partial \Phi^e}{\partial \mathbf{n}_y} - \Phi^e \frac{\partial u(y)}{\partial \mathbf{n}_y} \right\} ds(y) \qquad x \in \partial D, \tag{4.5b}$$

where the boundary condition on $u^t$ was used. A similar integral equation can be obtained for $\frac{\partial u}{\partial \mathbf{n}}$ by taking the normal derivative with respect to $x$ in equations (4.4). Using the boundary conditions and the jump conditions on the normal derivative of the potentials (the normal derivative of the double layer potential is continuous across the surface while the normal derivative of the single layer potential has a $+\frac{1}{2}u$ jump when

the limit is taken from the inside, and a $-\frac{1}{2}u$ jump when it is taken from the outside [46]), the following hold:

$$\frac{\nu}{2}\frac{\partial u(x)}{\partial \mathbf{n}_x} = \int_{\partial D}\left\{\nu\frac{\partial \Phi^i}{\partial \mathbf{n}_x}\frac{\partial u(y)}{\partial \mathbf{n}_y} - u(y)\frac{\partial^2 \Phi^i}{\partial \mathbf{n}_x \partial \mathbf{n}_y}\right\} ds(y) \qquad x \in \partial D, \tag{4.6a}$$

$$\frac{1}{2}\frac{\partial u(x)}{\partial \mathbf{n}_x} - \frac{\partial u^{inc}(x)}{\partial \mathbf{n}_x} = \int_{\partial D}\left\{u(y)\frac{\partial^2 \Phi^e}{\partial \mathbf{n}_x \partial \mathbf{n}_y} - \frac{\partial \Phi^e}{\partial \mathbf{n}_x}\frac{\partial u(y)}{\partial \mathbf{n}_y}\right\} ds(y) \qquad x \in \partial D. \tag{4.6b}$$

Note that one of the kernels in the previous equation is the double derivative of the Green's function, which has a non-integrable singularity as $y \to x$:

$$\frac{\partial^2 \Phi^e}{\partial \mathbf{n}_x \partial \mathbf{n}_y} \sim \frac{1}{|x-y|} \quad \text{in 2D}, \tag{4.7}$$

$$\frac{\partial^2 \Phi^e}{\partial \mathbf{n}_x \partial \mathbf{n}_y} \sim \frac{1}{|x-y|^2} \quad \text{in 3D}. \tag{4.8}$$

To avoid this singularity, it is more convenient to consider the difference of the Green's functions $\Phi^i - \Phi^e$. This is achieved by considering (4.5a)+(4.5b) and (4.6a)+(4.6b):

$$u(x) + \int_{\partial D}\left\{u(y)\frac{\partial(\Phi^i - \Phi^e)}{\partial \mathbf{n}_y} - \frac{\partial u(y)}{\partial \mathbf{n}_y}(\nu\Phi^i - \Phi^e)\right\} ds(y) = u^{inc}(x) \qquad x \in \partial D, \tag{4.9a}$$

$$\frac{1+\nu}{2}\frac{\partial u(x)}{\partial \mathbf{n}_x} + \int_{\partial D}\left\{u(y)\frac{\partial^2(\Phi^i - \Phi^e)}{\partial \mathbf{n}_x \partial \mathbf{n}_y} - \frac{\partial u(y)}{\partial \mathbf{n}_y}\frac{\partial(\nu\Phi^i - \Phi^e)}{\partial \mathbf{n}_x}\right\} ds(y) = \frac{\partial u^{inc}(x)}{\partial \mathbf{n}_x} \quad x \in \partial D. \tag{4.9b}$$

Note that the two integral equations are expressed as integrals over the infinite domain $\partial D$. This is inconvenient from a numerical point of view, but fortunately can be avoided by noticing that the boundary $\partial D$ is infinitely periodic with period $d$. Assuming the incident field is a monochromatic plane wave, it is quasi-periodic, and hence the solution $u$ must be quasi-periodic as well. Specifically, $u(x+d) = e^{i\alpha d}u(x)$. Hence, equations (4.5) and (4.6) can be rewritten over a single period $\mathbf{P}$ of the surface as

$$u(x) + \int_{\mathbf{P}}\left\{u(y)\frac{\partial(G^i - G^e)}{\partial \mathbf{n}_y} - \frac{\partial u(y)}{\partial \mathbf{n}_y}(\nu G^i - G^e)\right\} ds(y) = u^{inc}(x), \tag{4.10a}$$

$$\frac{1+\nu}{2}\frac{\partial u(x)}{\partial \mathbf{n}_x} + \int_{\mathbf{P}}\left\{u(y)\frac{\partial^2(G^i - G^e)}{\partial \mathbf{n}_x \partial \mathbf{n}_y} - \frac{\partial u(y)}{\partial \mathbf{n}_y}\frac{\partial(\nu G^i - G^e)}{\partial \mathbf{n}_x}\right\} ds(y) = \frac{\partial u^{inc}(x)}{\partial \mathbf{n}_x} \tag{4.10b}$$

for $x \in \mathbf{P}$, where

$$G(x,y) = \frac{i}{4} \sum_{m=-\infty}^{\infty} e^{i\alpha md} H_0^{(1)}(k|x - md\hat{x}_1 - y|) \quad \text{in 2D,} \tag{4.11}$$

$$G(x,y) = \frac{1}{4\pi} \sum_{m=-\infty}^{\infty} e^{i\alpha md} \frac{e^{ik|x-md\hat{x}_1-y|}}{|x - md\hat{x}_1 - y|} \quad \text{in 3D} \tag{4.12}$$

are the quasi-periodic Green's functions in 2D and 3D respectively. Hence all the kernels are weakly singular in both 2D and 3D (as shall be shown in the next section) and thus all the integrals exist.

The system (4.10) above can be viewed as an integral operator acting on the pair $[u; \frac{\partial u}{\partial \mathbf{n}}]$:

$$\mathcal{L} \begin{bmatrix} u \\ \frac{\partial u}{\partial \mathbf{n}} \end{bmatrix} = \begin{bmatrix} u^{inc} \\ \frac{\partial u^{inc}}{\partial \mathbf{n}} \end{bmatrix},$$

where $\mathcal{L}$ is the integral operator on the left hand side of (4.10). Thus, from Fredholm theory the system has a unique solution provided that the operator is compact, and that the only solution of the homogeneous system be the trivial one, which is true here because in the absence of incident fields, there can be neither scattered nor interior fields. Compactness of the integral operator follows from the assumption that the boundary $\partial D$ is twice differentiable, and the fact that the kernels are weakly singular [62].

## 4.2 Cylindrical domains

The integral equations derived in the previous section hold for any domain $D$ with $C^2$ boundary $\partial D$ and any incident field $u^{inc}$. In this section, the focus is restricted to two dimensional problems, that is to scattering from infinitely long cylindrical domains.

Consider a time-harmonic plane wave

$$u^{inc}(x) = e^{i(\alpha x_1 - \beta x_2)},$$

defined for every $x = (x_1, x_2) \in \mathbb{R}^2$ and incident at an angle $\theta$ from the horizontal, as shown in figure 4.1. Thus $\alpha = k_e \sin(\theta)$ and $\beta = k_e \cos(\theta)$, where $k_e$ is the wave number. This incident field is scattered by a periodic metallic grating with period $d$, whose surface

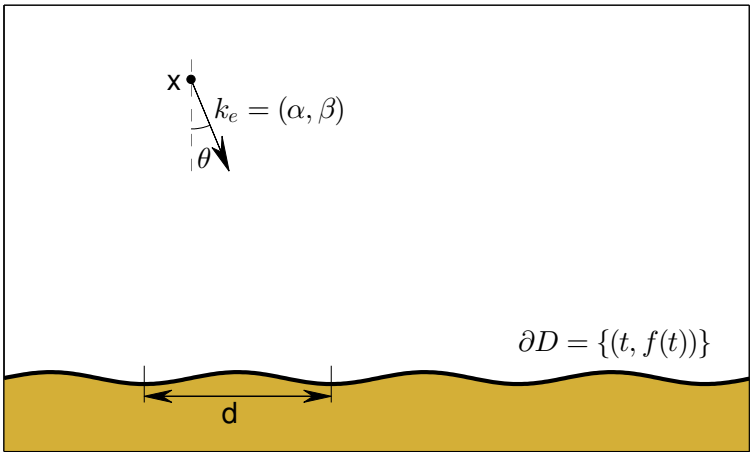



Figure 4.1: A 2D periodic metal grating is illuminated by an incident field.

is described by

$$\partial D = \left\{ y \in \mathbb{R}^2 \,:\, y = (t, f(t)) \right\},$$

where $f : \mathbb{R} \to \mathbb{R}$ is a smooth periodic function with $f(t + d) = f(t)$. The normal vector to this surface is given by

$$\mathbf{n}(t) = (n_1, n_2) = \frac{(-f'(t), 1)}{|y'(t)|}, \qquad |y'(t)| = \sqrt{1 + f'(t)^2}.$$

In order to parametrize the system of integral equations (4.10), choose the interval of integration $\mathbf{P} = \{y = (\tau, f(\tau)) \,:\, \tau \in [-d/2, d/2]\}$. The limits of integration are chosen such that any potential singularities in the kernels, which occur at $\tau = t$ lies well inside the integration interval. This is done in order to make numerical evaluations easier. Given $x(t) = (t, f(t))$ also on the surface $\partial D$, the system (4.10) can be re-written for $t \in [-d/2, d/2]$ as

$$\phi_1(t) + \int_{t-d/2}^{t+d/2} \{\phi_1(\tau) K_1(t, \tau) - \phi_2(\tau) K_2(t, \tau)\} \, d\tau = \psi_1(t), \tag{4.13a}$$

$$\frac{1+\nu}{2}\phi_2(t) + \int_{t-d/2}^{t+d/2} \{\phi_1(\tau) K_3(t, \tau) - \phi_2(\tau) K_4(t, \tau)\} \, d\tau = \psi_2(t), \tag{4.13b}$$

where the following notation was introduced for the unknown densities and the source

terms

$$\phi_1(t) = u(x(t)), \qquad \phi_2(t) = \frac{\partial u(x(t))}{\partial \mathbf{n}_x}, \tag{4.14a}$$

$$\psi_1(t) = u^{inc}(x(t)), \qquad \psi_2(t) = \frac{\partial u^{inc}(x(t))}{\partial \mathbf{n}_x}, \tag{4.14b}$$

and the new kernels are given by

$$K_1(t,\tau) = |y'(\tau)|\frac{\partial (G^i - G^e)}{\partial \mathbf{n}_y}(x(t), y(\tau)), \tag{4.15a}$$

$$K_2(t,\tau) = |y'(\tau)|(\nu G^i(x(t), y(\tau)) - G^e(x(t), y(\tau))), \tag{4.15b}$$

$$K_3(t,\tau) = |y'(\tau)|\frac{\partial^2 (G^i - G^e)}{\partial \mathbf{n}_x \partial \mathbf{n}_y}(x(t), y(\tau)), \tag{4.15c}$$

$$K_4(t,\tau) = |y'(\tau)|\frac{\partial (\nu G^i - G^e)}{\partial \mathbf{n}_x}(x(t), y(\tau)). \tag{4.15d}$$

Solving this system of integral equations for the unknown densities $\phi_1$ and $\phi_2$ will be the focus of the remaining sections in this chapter.

Once these quantities are known, recovering the transmitted and scattered fields away from the surface can be recovered from equations (4.4), which become

$$u^t(x) = \int_{-d/2}^{d/2} \left\{ \nu \phi_2(\tau) G^i(x, y(\tau)) - \phi_1(\tau) \frac{\partial G^i(x, y(\tau))}{\partial \mathbf{n}_y} \right\} |y'(\tau)| d\tau \qquad x \in D,$$

$$u^s(x) = \int_{-d/2}^{d/2} \left\{ \phi_1(\tau) \frac{\partial G^e(x, y(\tau))}{\partial \mathbf{n}_y} - \phi_2(\tau) G^e(x, y(\tau)) \right\} |y'(\tau)| d\tau \qquad x \in D^C.$$

using the surface parametrization introduced above. Computing the Green's functions $G^i$, $G^e$ and their normal derivatives and evaluating the integrals is significantly easier for $x \notin \partial D$, though numerical issues may still arise for $x$ close to the surface $\partial D$. For this reason, the same numerical techniques will be used to evaluate these integrals near the surface as will be used for solving the integral system (4.13).

For values of $x$ away from the surface on the other hand, the field can be computed by using the spectral series representation of the Green's functions

$$G^i(x,y) = \frac{i}{2d} \sum_{n=-\infty}^{\infty} \frac{e^{i\alpha_n(x_1 - y_1) + i\beta_n^i |x_2 - y_2|}}{\beta_n^i}, \tag{4.16a}$$

$$G^e(x,y) = \frac{i}{2d} \sum_{n=-\infty}^{\infty} \frac{e^{i\alpha_n(x_1 - y_1) + i\beta_n^e |x_2 - y_2|}}{\beta_n^e}, \tag{4.16b}$$

which converge uniformly on compact sets where $x_2 \neq y_2$ [16], and where $\alpha_n = \alpha + 2n\pi/d$ and $\beta_n^{i,e} = \sqrt{k_{i,e}^2 - \alpha_n^2}$ for the interior and exterior Green's functions respectively. Thus for $x_2 > \max_{x_1} f(x_1)$, the scattered field above the surface can be obtained from the exponentially convergent Rayleigh series

$$u^s(x_1, x_2) = \sum_{n=-\infty}^{\infty} b_n^e e^{i(\alpha_n x_1 + \beta_n^e x_2)}, \tag{4.17}$$

where the Rayleigh coefficients $b_n^e$ are given by

$$b_n^e = \frac{1}{2d\beta_n^e} \int_{t-d/2}^{t+d/2} e^{-i(\alpha_n \tau + \beta_n^e f(\tau))} \left[ \phi_1(\tau)(\beta_n^e - \alpha_n f'(\tau)) - i\phi_2(\tau)\sqrt{1 + f'(\tau)^2} \right] d\tau. \tag{4.18}$$

Similarly, the transmitted field below the surface can be obtained by means of a Rayleigh series which converges for $x_2 < \min_{x_1} f(x_1)$

$$u^t(x_1, x_2) = \sum_{n=-\infty}^{\infty} b_n^i e^{i(\alpha_n x_1 - \beta_n^i x_2)}, \tag{4.19}$$

where the coefficients are

$$b_n^i = \frac{1}{2d\beta_n^i} \int_{t-d/2}^{t+d/2} e^{-i(\alpha_n \tau - \beta_n^i f(\tau))} \left[ \phi_1(\tau)(\beta_n^i + \alpha_n f'(\tau)) + i\nu\phi_2(\tau)\sqrt{1 + f'(\tau)^2} \right] d\tau. \tag{4.20}$$

Note the negative sign in the exponential terms in (4.19), which indicates that the transmitted field is propagating downward away from the surface.

## 4.3 Kernel decompositions

In this section, the singularities in the kernels from equations (4.13) are shown explicitly in 2D. Recall that the quasi-periodic Green's function is given by

$$G(x, y) = \frac{i}{4} \sum_{m=-\infty}^{\infty} e^{i\alpha md} H_0^{(1)}(k|x - md\hat{x}_1 - y|). \tag{4.21}$$

Thus, for $\tau \in [t - d/2, t + d/2]$, $G(x(t), y(\tau))$ has only one singularity, and can be decomposed as follows:

$$G(x, y) = \frac{i}{4} H_0^{(1)}(k|x - y|) + \mathcal{D}(x, y), \tag{4.22}$$

where $\mathcal{D}$ is an analytic function of its arguments [52].

$$\frac{\partial}{\partial \mathbf{n}(\mathbf{r}')} G(k|\mathbf{r}-\mathbf{r}'|) = kG'(k|\mathbf{r}-\mathbf{r}'|)\frac{c(x,x')}{|\mathbf{r}-\mathbf{r}'|\sqrt{1+f'(x')^2}} \tag{4.23}$$

$$\frac{\partial^2}{\partial \mathbf{n}(\mathbf{r})\partial \mathbf{n}(\mathbf{r}')} G(k|\mathbf{r}-\mathbf{r}'|) = k^2 G''(k|\mathbf{r}-\mathbf{r}'|)\frac{c(x,x')c(x',x)}{|\mathbf{r}-\mathbf{r}'|^2\sqrt{1+f'(x')^2}\sqrt{1+f'(x)^2}} \tag{4.24}$$

$$- kG'(k|\mathbf{r}-\mathbf{r}'|)\frac{c(x,x')c(x',x)}{|\mathbf{r}-\mathbf{r}'|^3\sqrt{1+f'(x')^2}\sqrt{1+f'(x)^2}} \tag{4.25}$$

$$- kG'(k|\mathbf{r}-\mathbf{r}'|)\frac{f'(x)f'(x')+1}{|\mathbf{r}-\mathbf{r}'|\sqrt{1+f'(x')^2}\sqrt{1+f'(x)^2}} \tag{4.26}$$

## 4.4 Numerical solution

This section gives some details about the way the Nystrom method is used to discretize the integral system (4.10) and the solution $u(x)$ and $\frac{\partial u(x)}{\partial \mathbf{n}}$ is obtained for $x \in \partial D$.

A truncated Floquet expansion with $N$ terms is used to approximate $\phi_1$ and $\phi_2$, which is known to converge superalgebraically [52]:

$$\phi_1^N(t) = \sum_{n=-N}^{N} a_n e^{i\alpha_n t} \qquad\qquad \phi_2^N(t) = \sum_{n=-N}^{N} b_n e^{i\alpha_n t} \tag{4.27}$$

where $\alpha_n = \alpha + 2n\pi/d$ and $\beta_n = \beta_n^e = \sqrt{k_e^2 - \alpha_n^2}$ as before. With this approximation, the integral system (4.13) becomes

$$\psi_1(t) = \sum_{n=-N}^{N} a_n \left(e^{i\alpha_n t} + W_n^{K_1}(t)\right) - b_n W_n^{K_2}(t), \tag{4.28a}$$

$$\psi_2(t) = \sum_{n=-N}^{N} a_n W_n^{K_3} + b_n \left(\frac{1+\nu}{2} e^{i\alpha_n t} - W_n^{K_4}\right) \tag{4.28b}$$

where the weights $W_n$ are given by

$$W_n^{K_j}(t) = \int_{-d/2}^{d/2} K_j(t,\tau) e^{i\alpha_n \tau} d\tau. \tag{4.29}$$

At this point, the integral system (4.13) has been reduced to a linear system (4.28) of equations for the Floquet coefficients $a_n$ and $b_n$. To solve it, a collocation method is used, in which equations (4.28) are evaluated explicitly at $2N+1$ equally spaced values

of $t$, thus resulting in a linear system of $4N + 2$ equations in $4N + 2$ unknowns, which can then be solved via the usual methods.

Once the coefficients $a_n$ and $b_n$ are solved for, the values of the field $u(x)$ and $\frac{\partial u(x)}{\partial \mathbf{n}}$ for $x \in \partial D$ can be obtained from equations (4.27), recalling the notation $u((t, f(t)) = \phi_1(t)$ and $\frac{\partial u}{\partial \mathbf{n}}((t, f(t))) = \phi_2(t)$.

## 4.5 Numerical results

The algorithm described has been implemented in Matlab. The test case used in the simulation is a metal grating, extending infinitely in one direction and infinitely periodic in the perpendicular direction. The parameters used are in the plasmonic regime, that is the height $h$, period $d$ and wavelength $\lambda$ of incident light are such that

$$\frac{\lambda}{d} \approx 1 \qquad\qquad \frac{\lambda}{h} \approx 0.01. \tag{4.30}$$

In particular, the profile of the grating is given by

$$f(x) = \frac{h}{2} \sin(x \frac{2\pi}{d}), \tag{4.31}$$

where $h = 24$nm, $d = 300$nm and the dielectric constant of the metal is given by $\epsilon_{metal} = -3.304 + 0.577i$, while that of the material above the grating is taken to be 1. The incoming plane wave is has incident angle $\theta = 27^\circ$ and wavelength $\lambda = 226$nm.

The first part of the algorithm solves for $u(x)$ and $\frac{\partial u(x)}{\partial \mathbf{n}}$, where the upward pointing normal was used for $x \in \partial D$. The results shown in figure 4.2 show that the field $u$ and its normal derivative $\frac{\partial u}{\partial \mathbf{n}}$ are smooth and well resolved with $N = 32$. Moreover, figure 4.3 shows the maximum error in both these quantities as a function of discretization number $N$. Note that on a log scale, the error decreases linearly, indicating the algorithm does indeed converge exponentially, as should be expected. The error was computed by comparing the results from one discretization with the next, i.e. $error(N) = \max_x |u^{2N}(x) - u^N(x)|$.

From the surface field, the field away from the surface can be computed by the Rayleigh series, and the results are shown in figure 4.4. The plasmonic effect is observed by noting that the field is confined in a region of about 100nm above the surface, a scale smaller than the wavelength.

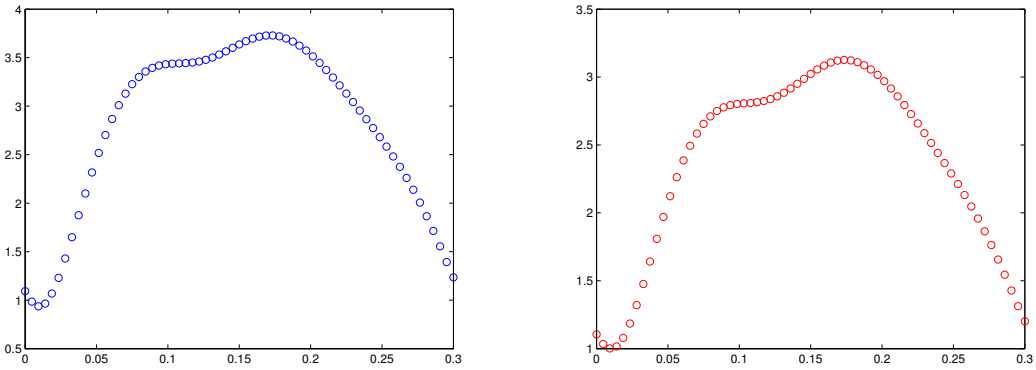



Figure 4.2: The total field (left) and the normal derivative (right) of the total field on the surface of the grating. Here $N = 32$.

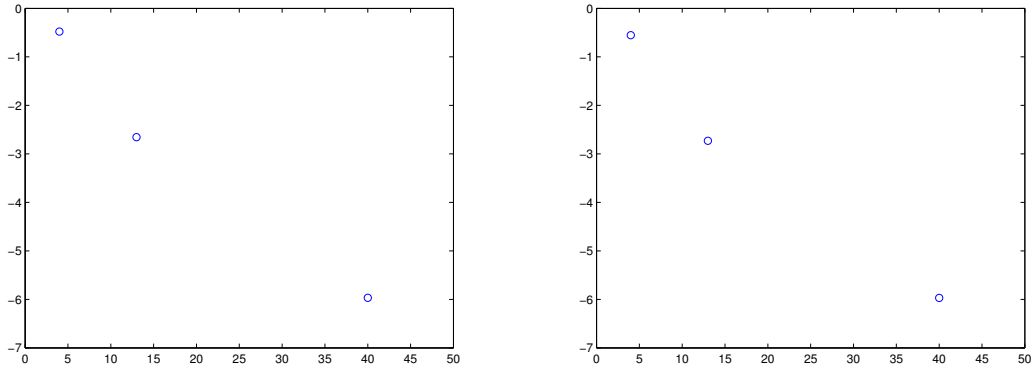



Figure 4.3: The error in the total field (left) and its normal derivative (right) as a function of the number of points. The error is shown on a logarithmic scale.

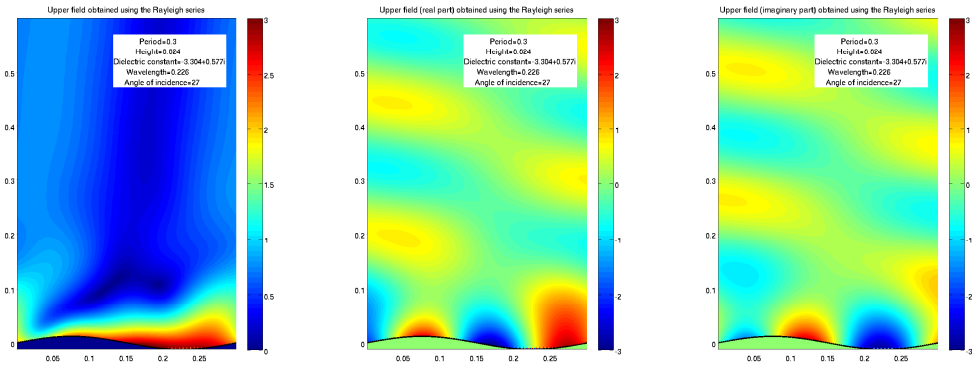



Figure 4.4: The field above the surface of the grating. From left to right, the norm of the field, the real part and the imaginary part.

# Chapter 5

# Shape derivatives

The solution $u$ to the scattering problem described in the previous chapter depends on the profile of the grating, which can be described in terms of parameters $\mathbf{p} = (p_1, p_2, ..., p_r)$. The dependence of the solution $u$ on these parameters is described by the gradient $\nabla_{\mathbf{p}} u$ in parameter space, which can be obtained at little additional cost by the same numerical solver that yields the surface field $u$ itself, as shall be described in this chapter.

## 5.1 Differential formulation

In this section, a system of differential equations is derived for the gradient $\nabla_{\mathbf{p}} u$ of the field $u$ in the space of parameters $\mathbf{p} \in \mathbb{R}^r$. It is shown that this system is the same as the original system (2.6a) with new boundary conditions, and thus solving for $\nabla_{\mathbf{p}} u$ can be done using the same integral methods used in solving for $u$.

Consider $u_p = \frac{\partial u}{\partial p}$, the partial derivative of the field $u$ with respect to a scalar parameter $p$ of the surface profile. Recall that the scattered and transmitted fields $u^s$

and $u^t$ satisfy the system of helmholtz' equations (2.6a), thus:

$$\Delta u^t(x_1, x_2, p) + k_i^2 u^t(x_1, x_2, p) = 0 \qquad \text{on } D, \tag{2.6a}$$

$$\Delta u^s(x_1, x_2, p) + k_e^2 u^s(x_1, x_2, p) = 0 \qquad \text{on } D^C, \tag{2.6b}$$

$$\begin{aligned} u^t(x_1, x_2, p) &= u^s(x_1, x_2, p) + u^{inc}(x_1, x_2, p) \\ &\equiv u(x_1, x_2, p) \qquad \text{on } \partial D, \end{aligned} \tag{2.6d}$$

$$\frac{\partial u^t(x_1, x_2, p)}{\partial \mathbf{n}} = \nu \frac{\partial}{\partial \mathbf{n}} u(x_1, x_2, p) \qquad \text{on } \partial D. \tag{2.6e}$$

Since in equations (2.6a) and (2.6b) the only dependence on $p$ explicitly, the partial derivative with respect to $p$, may be applied to either side of both, yielding

$$\Delta u_p^t(x_1, x_2, p) + k_i^2 u_p^t(x_1, x_2, p) = 0 \qquad \text{on } D, \tag{5.2a}$$

$$\Delta u_p^s(x_1, x_2, p) + k_e^2 u_p^s(x_1, x_2, p) = 0 \qquad \text{on } D^C. \tag{5.2b}$$

Equations (2.6d) and (2.6e) on the other hand also have an implicit dependence on $p$, through the parametrization $(x_1, x_2) = (t, f(t, p))$, so a full derivative $\frac{d}{dp}$ must be taken in order to preserve the equality. Thus, from equation (2.6d) the following is obtained

$$u_p^t + f_p \frac{\partial u^t}{\partial x_2} = u_p + f_p \frac{\partial u}{\partial x_2},$$

while (2.6e) yields

$$\frac{d}{dp}\left\{\mathbf{n} \cdot \nabla u^t\right\} = \nu \frac{d}{dp}\left\{\mathbf{n} \cdot \nabla u\right\},$$

$$\frac{d\mathbf{n}}{dp} \cdot \nabla u^t + \mathbf{n} \cdot \nabla \frac{\partial u^t}{\partial p} + f_p \mathbf{n} \cdot \frac{\partial}{\partial x_2} \nabla u^t = \nu \left\{ \frac{d\mathbf{n}}{dp} \cdot \nabla u + \mathbf{n} \cdot \nabla \frac{\partial u}{\partial p} + f_p \mathbf{n} \cdot \frac{\partial}{\partial x_2} \nabla u \right\}.$$

The new boundary conditions may then be written as

$$u_p^t = u_p + \xi_1 \qquad \text{on } \partial D, \tag{5.2c}$$

$$\frac{\partial u_p^t}{\partial \mathbf{n}} = \nu \frac{\partial u_p}{\partial \mathbf{n}} + \xi_2 \qquad \text{on } \partial D, \tag{5.2d}$$

where

$$\xi_1 = f_p \frac{\partial u}{\partial x_2} - f_p \frac{\partial u^t}{\partial x_2}, \tag{5.3}$$

$$\xi_2 = \frac{d\mathbf{n}}{dp} \cdot \left(\nu \nabla u - \nabla u^t\right) + f_p \left(\nu \mathbf{n} \cdot \frac{\partial}{\partial x_2} \nabla u - \mathbf{n} \cdot \frac{\partial}{\partial x_2} \nabla u^t\right). \tag{5.4}$$

Note that while the fields $u^t$ and $u^s$ are known on the surface $\partial D$, their derivatives in $x_2$ are not. However, their normal derivatives are known, and the following change of variables provides a way of going from the coordinate system $(x_1, x_2)$ to the local coordinate system $(\mathbf{n}, \mathbf{s})$, where $\mathbf{n}$ is the unit normal to the surface while $\mathbf{s}$ is the unit tangent:

$$\frac{\partial}{\partial \mathbf{n}} = n_1 \frac{\partial}{\partial x_1} + n_2 \frac{\partial}{\partial x_2} \qquad \frac{\partial}{\partial \mathbf{s}} = n_2 \frac{\partial}{\partial x_1} - n_1 \frac{\partial}{\partial x_2}$$
$$\frac{\partial}{\partial x_1} = n_2 \frac{\partial}{\partial \mathbf{s}} + n_1 \frac{\partial}{\partial \mathbf{n}} \qquad \frac{\partial}{\partial x_2} = -n_1 \frac{\partial}{\partial \mathbf{s}} + n_2 \frac{\partial}{\partial \mathbf{n}}.$$

Combining these derivatives an expression for the second derivatives appearing in (5.4) can be obtained:

$$\frac{\partial^2}{\partial x_1 \partial x_2} = -n_2 \frac{\partial n_1}{\partial \mathbf{s}} \frac{\partial}{\partial \mathbf{s}} + n_2 \frac{\partial n_2}{\partial \mathbf{s}} \frac{\partial}{\partial \mathbf{n}} - n_1 n_2 \frac{\partial^2}{\partial \mathbf{s}^2} + n_1 n_2 \frac{\partial^2}{\partial \mathbf{n}^2} + (n_2^2 - n_1^2) \frac{\partial^2}{\partial \mathbf{s} \partial \mathbf{n}},$$
$$\frac{\partial^2}{\partial x_2^2} = n_1 \frac{\partial n_1}{\partial \mathbf{s}} \frac{\partial}{\partial \mathbf{s}} - n_1 \frac{\partial n_2}{\partial \mathbf{s}} \frac{\partial}{\partial \mathbf{n}} + n_1^2 \frac{\partial^2}{\partial \mathbf{s}^2} + n_2^2 \frac{\partial^2}{\partial \mathbf{n}^2} - 2 n_1 n_2 \frac{\partial^2}{\partial \mathbf{s} \partial \mathbf{n}},$$
$$\mathbf{n} \cdot \frac{\partial}{\partial x_2} \nabla = n_1 \frac{\partial^2}{\partial x_1 \partial x_2} + n_2 \frac{\partial^2}{\partial x_2^2} = n_2 \frac{\partial^2}{\partial \mathbf{n}^2} - n_1 \frac{\partial^2}{\partial \mathbf{s} \partial \mathbf{n}}.$$

Using these relations, equations (5.3) and (5.4) become

$$\xi_1 = -f_p n_1 \left( \frac{\partial u}{\partial \mathbf{s}} - \frac{\partial u^t}{\partial \mathbf{s}} \right) + f_p n_2 \left( \frac{\partial u}{\partial \mathbf{n}} - \frac{\partial u^t}{\partial \mathbf{n}} \right),$$
$$\xi_2 = \frac{d\mathbf{n}}{dp} \cdot \left\{ \nu \left( n_2 \frac{\partial u}{\partial \mathbf{s}} + n_1 \frac{\partial u}{\partial \mathbf{n}} \right) - n_2 \frac{\partial u^t}{\partial \mathbf{s}} - n_1 \frac{\partial u^t}{\partial \mathbf{n}} \right.,$$
$$\left. \nu \left( -n_1 \frac{\partial u}{\partial \mathbf{s}} + n_2 \frac{\partial u}{\partial \mathbf{n}} \right) + n_1 \frac{\partial u^t}{\partial \mathbf{s}} - n_2 \frac{\partial u^t}{\partial \mathbf{n}} \right\}$$
$$+ f_p \left\{ \nu \left( n_2 \frac{\partial^2}{\partial \mathbf{n}^2} - n_1 \frac{\partial^2}{\partial \mathbf{s} \partial \mathbf{n}} \right) u - \left( n_2 \frac{\partial^2}{\partial \mathbf{n}^2} - n_1 \frac{\partial^2}{\partial \mathbf{s} \partial \mathbf{n}} \right) u^t \right\}.$$

From the boundary conditions, $u^t = u$ and $\frac{\partial u^t}{\partial \mathbf{n}} = \nu \frac{\partial u}{\partial \mathbf{n}}$ on $\partial D$, thus

$$\xi_1 = f_p n_2 (1 - \nu) \frac{\partial u}{\partial \mathbf{n}}, \tag{5.5}$$
$$\xi_2 = (\nu - 1) \frac{d\mathbf{n}}{dp} \cdot (n_2, -n_1) \frac{\partial u}{\partial \mathbf{s}} + n_2 f_p \left( \nu \frac{\partial^2 u}{\partial \mathbf{n}^2} - \frac{\partial^2 u^t}{\partial \mathbf{n}^2} \right). \tag{5.6}$$

The second normal derivative can be recovered from the Helmholtz equation $\Delta u = -k^2 u$ by noticing that the Laplace operator can be written as $\Delta = \frac{\partial^2}{\partial \mathbf{n}^2} + \frac{\partial^2}{\partial \mathbf{s}^2}$. Using this fact,

together with the boundary condition $u = u^t$,

$$\xi_2 = (\nu - 1)\frac{d\mathbf{n}}{dp} \cdot (n_2, -n_1)\frac{\partial u}{\partial \mathbf{s}} + n_2 f_p \left( (k_i^2 - \nu k_e^2)u + (1 - \nu)\frac{\partial^2 u}{\partial \mathbf{s}^2} \right). \tag{5.7}$$

Since $\nu = \frac{\epsilon_i}{\epsilon_e} = \frac{k_i^2}{k_e^2}$, the $u$ term drops out, so

$$\xi_2 = (\nu - 1)\frac{d\mathbf{n}}{dp} \cdot (n_2, -n_1)\frac{\partial u}{\partial \mathbf{s}} + n_2 f_p (1 - \nu)\frac{\partial^2 u}{\partial \mathbf{s}^2}. \tag{5.8}$$

Recalling further that the parametrization $(x_1, x_2) = (t, f(t, p))$ has an upward normal given by $n_1 = \frac{-f_t(t,p)}{\sqrt{1+f_t(t,p)^2}}$, $n_2 = \frac{1}{\sqrt{1+f_t(t,p)^2}}$, then the derivatives with respect to p are given by

$$\frac{d\mathbf{n}}{dp} = \frac{(-1, -f_t(t, p))}{(1 + f_t(t, p)^2)^{3/2}} f_{tp}(t, p) = n_2^2 f_{tp}(-n_2, n_1).$$

Thus the expressions for $\xi_1$ and $\xi_2$ can be simplified to

$$\xi_1 = f_p n_2 (1 - \nu)\frac{\partial u}{\partial \mathbf{n}}, \tag{5.9}$$

$$\xi_2 = n_2^2 f_{tp}(1 - \nu)\frac{\partial u}{\partial \mathbf{s}} + n_2 f_p (1 - \nu)\frac{\partial^2 u}{\partial \mathbf{s}^2}. \tag{5.10}$$

At this point, all quantities are known: $f$ is prescribed, $u$ and $\frac{\partial u}{\partial \mathbf{n}}$ on the surface are obtained from the integral solver. The tangential derivatives can be recast as derivatives in the parameter $t$ by noticing that

$$\begin{aligned}
\frac{\partial u}{\partial \mathbf{s}} &= n_2 \frac{\partial u}{\partial x_1} - n_1 \frac{\partial u}{\partial x_2} = n_2 \frac{\partial u}{\partial x_1} + n_2 f_t \frac{\partial u}{\partial x_2} = n_2 \left( \frac{\partial u}{\partial x_1} + \frac{\partial u}{\partial x_2} f_t \right) = n_2 \frac{du}{dt}, \\
\frac{\partial^2 u}{\partial \mathbf{s}^2} &= n_2 \frac{d}{dt}\left( n_2 \frac{du}{dt} \right) = n_2 \left( \frac{dn_2}{dt}\frac{du}{dt} + n_2 \frac{d^2 u}{dt^2} \right) = n_2 \left( n_1 n_2^2 f_{tt}\frac{du}{dt} + n_2 \frac{d^2 u}{dt^2} \right).
\end{aligned}$$

To summarize, the dependence of the scattering solution $u$ on a profile parameter $p$ satisfies the following PDE system:

$$\Delta u_p^t + k_i^2 u_p^t = 0 \qquad \text{on } D, \tag{5.11a}$$

$$\Delta u_p + k_e^2 u_p = 0 \qquad \text{on } D^C, \tag{5.11b}$$

$$u_p^t = u_p + \xi_1 \qquad \text{on } \partial D, \tag{5.11c}$$

$$\frac{\partial u_p^t}{\partial \mathbf{n}} = \nu \frac{\partial u_p}{\partial \mathbf{n}} + \xi_2 \qquad \text{on } \partial D, \tag{5.11d}$$

where

$$\xi_1 = f_p n_2 (1-\nu)\frac{\partial u}{\partial \mathbf{n}}, \tag{5.12}$$

$$\xi_2 = n_2^3 (1-\nu)\left( (f_{tp} + f_p f_{tt} n_1 n_2)\frac{du}{dt} + f_p \frac{d^2 u}{dt^2} \right), \tag{5.13}$$

and $\mathbf{n} = (n_1, n_2)$ is the upward pointing unit normal to the surface $\partial D$. Note that all the quantities used in the formula for $\xi_1$ and $\xi_2$ are known, except for the derivatives of $u$ with respect to the parameter $t$, which can be easily computed via Fourier transforms, since $u$ is quasi-periodic.

## 5.2 Integral formulation

In this section, a system of integral equations is derived for $u_p$. This system is the same as the integral system (4.10) with a new right-hand side, and thus solving for $u_p$ will have the same computational cost as solving for $u$ once this new right hand side is computed.

From the system of PDE's (5.11), expressions similar to (4.5) are obtained for $u_p^i$ and $u_p$ using Green's theorem and taking limits as $x \to \partial D$

$$\frac{1}{2}u_p^i(x) = \int_{\partial D} \left\{ \Phi^i(x,y)\frac{\partial u_p^i(y)}{\partial \mathbf{n}_y} - u_p^i(y)\frac{\partial \Phi^i(x,y)}{\partial \mathbf{n}_y} \right\} ds(y) \qquad x \in \partial D, \tag{5.14a}$$

$$\frac{1}{2}u_p(x) = \int_{\partial D} \left\{ u_p(y)\frac{\partial \Phi^e(x,y)}{\partial \mathbf{n}_y} - \Phi^e(x,y)\frac{\partial u_p(y)}{\partial \mathbf{n}_y} \right\} ds(y) \qquad x \in \partial D. \tag{5.14b}$$

Using the boundary conditions (5.11c) on $u_p^t$ and (5.11d) on $\frac{\partial u_p^t}{\partial \mathbf{n}}$,

$$\frac{1}{2}[u_p(x) + \xi_1(x)] = \int_{\partial D} \left\{ \Phi^i(x,y)\left[ \nu \frac{\partial u_p(y)}{\partial \mathbf{n}_y} + \xi_2(y) \right] - [u_p(y) + \xi_1(y)]\frac{\partial \Phi^i(x,y)}{\partial \mathbf{n}_y} \right\} ds(y) \qquad x \in \partial D, \tag{5.15a}$$

$$\frac{1}{2}u_p(x) = \int_{\partial D} \left\{ u_p(y)\frac{\partial \Phi^e(x,y)}{\partial \mathbf{n}_y} - \Phi^e(x,y)\frac{\partial u_p(y)}{\partial \mathbf{n}_y} \right\} ds(y) \quad x \in \partial D. \tag{5.15b}$$

Expressions similar to (4.6) are obtained for $\frac{\partial u_p^t}{\partial \mathbf{n}}$ and $\frac{\partial u_p}{\partial \mathbf{n}}$ by taking the normal

derivative with respect to $x$ in Green's theorem and letting $x \to \partial D$:

$$\frac{1}{2}\frac{\partial u_p^t(x)}{\partial \mathbf{n}_x} = \int_{\partial D}\left\{\frac{\partial \Phi^i(x,y)}{\partial \mathbf{n}_x}\frac{\partial u_p^t(y)}{\partial \mathbf{n}_y} - u_p^t(y)\frac{\partial^2 \Phi^i(x,y)}{\partial \mathbf{n}_x \partial \mathbf{n}_y}\right\} ds(y) \qquad x \in \partial D, \tag{5.16a}$$

$$\frac{1}{2}\frac{\partial u_p(x)}{\partial \mathbf{n}_x} = \int_{\partial D}\left\{u_p(y)\frac{\partial^2 \Phi^e(x,y)}{\partial \mathbf{n}_x \partial \mathbf{n}_y} - \frac{\partial \Phi^e(x,y)}{\partial \mathbf{n}_x}\frac{\partial u_p(y)}{\partial \mathbf{n}_y}\right\} ds(y) \qquad x \in \partial D. \tag{5.16b}$$

Using the boundary conditions (5.11c) on $u_p^t$ and (5.11d) on $\frac{\partial u_p^t}{\partial \mathbf{n}}$,

$$\frac{1}{2}\left[\nu\frac{\partial u_p(x)}{\partial \mathbf{n}_x} + \xi_2(x)\right] = \int_{\partial D}\left\{\frac{\partial \Phi^i(x,y)}{\partial \mathbf{n}_x}\left[\nu\frac{\partial u_p}{\partial \mathbf{n}_y} + \xi_2\right] - [u_p + \xi_1]\frac{\partial^2 \Phi^i(x,y)}{\partial \mathbf{n}_x \partial \mathbf{n}_y}\right\} ds(y) \qquad x \in \partial D, \tag{5.17a}$$

$$\frac{1}{2}\frac{\partial u_p(x)}{\partial \mathbf{n}_x} = \int_{\partial D}\left\{u_p\frac{\partial^2 \Phi^e(x,y)}{\partial \mathbf{n}_x \partial \mathbf{n}_y} - \frac{\partial \Phi^e(x,y)}{\partial \mathbf{n}_x}\frac{\partial u_p}{\partial \mathbf{n}_y}\right\} ds(y) \quad x \in \partial D. \tag{5.17b}$$

Adding equations (5.15) together, and equations (5.17) together, and taking advantage of the quasi-periodicity of the unknowns $u_p$ and $\frac{\partial u_p}{\partial \mathbf{n}}$ as well as that of the auxiliary quantities $\xi_1$ and $\xi_2$,

$$u_p + \int_{\mathbf{P}}\left\{u_p\frac{\partial (G^i - G^e)}{\partial \mathbf{n}_y} - (\nu G^i - G^e)\frac{\partial u_p}{\partial \mathbf{n}_y}\right\} ds(y) = -\frac{1}{2}\xi_1 + \int_{\mathbf{P}}\left\{G^i\xi_2 - \xi_1\frac{\partial G^i}{\partial \mathbf{n}_y}\right\} ds(y), \tag{5.18a}$$

$$\frac{\nu+1}{2}\frac{\partial u_p}{\partial \mathbf{n}_x} + \int_{\mathbf{P}}\left\{u_p\frac{\partial^2 (G^i - G^e)}{\partial \mathbf{n}_x \partial \mathbf{n}_y} - \frac{\partial (\nu G^i - G^e)}{\partial \mathbf{n}_x}\frac{\partial u_p}{\partial \mathbf{n}_y}\right\} ds(y) = -\frac{1}{2}\xi_2 + \int_{\mathbf{P}}\left\{\frac{\partial G^i}{\partial \mathbf{n}_x}\xi_2 - \xi_1\frac{\partial^2 G^i}{\partial \mathbf{n}_x \partial \mathbf{n}_y}\right\} ds(y). \tag{5.18b}$$

Note the integral operator on the left-hand side of the system is the same as in system (4.10), so it can be discretized and inverted numerically in the same way. The system can be thus expressed compactly as

$$\mathcal{L}\begin{bmatrix} u_p \\ \frac{\partial u_p}{\partial \mathbf{n}} \end{bmatrix} = rhs,$$

where $\mathcal{L}$ is the integral operator on the left hand side of (5.18), and $rhs$ is its right hand side. The only difference between this system and (4.10) is this new right-hand side.

While most of the integral terms appearing here are similar to the ones appearing in the operator $\mathcal{L}$ and can be evaluated using similar numerical techniques, the normal derivative of the double-layer potential, $T\xi_1$ is fundamentally different:

$$\frac{1}{2}(T\xi_1)(x) = \int_{\mathbf{P}} \xi_1(y) \frac{\partial^2 G^i(x,y)}{\partial \mathbf{n}_x \partial \mathbf{n}_y} ds(y), \quad x \in \partial D. \tag{5.19}$$

This integral operator can be handled using the identity

$$T\xi_1 = \frac{d}{ds} S\left(\frac{d\xi_1}{ds}\right) + k_i^2 \mathbf{n} \cdot S(\mathbf{n}\xi_1), \tag{5.20}$$

which holds as long as $\xi_1 \in C^{1,\eta}$ for $0 < \eta < 1$ [63], and where $s$ is an arclength parametrization of the surface and $S\xi_1$ is the single-layer potential

$$\frac{1}{2}(S\xi_1)(x) = \int_{\mathbf{P}} G^i(x,y)\xi_1(y) ds(y). \tag{5.21}$$

Thus, given the parametrization $x = x(t) = (t, f(t))$, $y = y(\tau) = (\tau, f(\tau))$,

$$\begin{aligned} \frac{(T\xi_1)(x(t))}{2} =& \frac{1}{|x'(t)|} \frac{d}{dt} \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} G^i(x(t), y(\tau)) \xi_1'(\tau) d\tau \\ &+ k_i^2 \mathbf{n}(t) \cdot \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} G^i(x(t), y(\tau)) \mathbf{n}(\tau) \xi_1(\tau) |y'(\tau)| d\tau. \end{aligned} \tag{5.22}$$

Since the integrand $G^i(x(t), y(\tau))\xi_1'(\tau)$ is periodic, applying Leibniz's integral rule we can take the derivative $\frac{d}{dt}$ inside the integral:

$$\begin{aligned} \frac{(T\xi_1)(x(t))}{2} =& \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} \frac{1}{|x'(t)|} \frac{d}{dt} G^i(x(t), y(\tau)) \xi_1'(\tau) d\tau \\ &+ k_i^2 \mathbf{n}(t) \cdot \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} G^i(x(t), y(\tau)) |y'(\tau)| \mathbf{n}(\tau) \xi_1(\tau) d\tau. \end{aligned} \tag{5.23}$$

Thus the right hand side of system (5.18) can be expressed as

$$rhs_1 = -\frac{1}{2}\xi_1(t) + I_5(t) - I_6(t), \tag{5.24a}$$

$$rhs_2 = -\frac{1}{2}\xi_2(t) + I_7(t) - I_8(t) - I_9(t), \tag{5.24b}$$

where using the same surface parametrization as before,

$$I_5(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} \underbrace{G^i(x(t), y(\tau))|y'(\tau)|}_{K_5(t,\tau)} \xi_2(\tau) d\tau, \tag{5.25a}$$

$$I_6(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} \underbrace{\frac{\partial G^i(x(t), y(\tau))}{\partial \mathbf{n}_y}|y'(\tau)|}_{K_6(t,\tau)} \xi_1(\tau) d\tau, \tag{5.25b}$$

$$I_7(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} \underbrace{\frac{\partial G^i(x(t), y(\tau))}{\partial \mathbf{n}_x}|y'(\tau)|}_{K_7(t,\tau)} \xi_2(\tau) d\tau, \tag{5.25c}$$

$$I_8(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} \underbrace{\frac{1}{|x'(t)|}\frac{d}{dt} G^i(x(t), y(\tau))}_{K_8(t,\tau)} \xi_1'(\tau) d\tau. \tag{5.25d}$$

$$I_9(t) = k_i^2 \mathbf{n}(t) \cdot \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} \underbrace{G^i(x(t), y(\tau))|y'(\tau)|}_{K_5(t,\tau)} \mathbf{n}(\tau) \xi_1(\tau) d\tau, \tag{5.25e}$$

Note that kernels $K_5$, $K_6$ and $K_7$ are weakly singular, and can be evaluated using largely the same decomposition and Chebyshev quadrature that was used for the weakly singular kernels on the left-hand-side. As for the hypersingular kernel $K_8$, it is possible to integrate it efficiently using a different decomposition and exact evaluation of a complex contour integral, as shall be discussed in the following sections.

## 5.3 Decomposition of the weakly singular kernels $K_5$, $K_6$ and $K_7$

In this section we show that each of the kernels $K_5$, $K_6$ and $K_7$ has a logarithmic singularity and can be decomposed as $K(t,\tau) = K_{,1}(t,\tau)\ln|t-\tau| + K_{,2}(t,\tau)$, where $K_{,1}(t,\tau)$ and $K_{,2}(t,\tau)$ are smooth functions of their arguments.

Recall that for points $x$, $y$ within one period of each other, the Green's function may be decomposed as follows, (4.22)

$$G(x,y) = \frac{i}{4}H_0^{(1)}(k|x-y|) + \mathcal{D}(x,y), \tag{5.26}$$

where $\mathcal{D}(x,y)$ is an analytic function of its arguments. Let $r(t,\tau) = |x(t) - y(\tau)| = \sqrt{(t-\tau)^2 + (f(t)-f(\tau))^2}$. Using this decomposition of the Green's function, together

with the series representation (A.7a) of the Hankel function of first kind, the logarithmic singularity of the $K_5(t,\tau)$ kernel can be isolated as follows:

$$\begin{aligned}
K_5(t,\tau) &= G^i(x(t),y(\tau))|y'(\tau)| \\
&= \left[\frac{i}{4}H_0^{(1)}(k_i r(t,\tau)) + \mathcal{D}(x,y)\right]|y'(\tau)| \\
&= \frac{i}{4}\left[\frac{2i}{\pi}J_0(k_i r(t,\tau))\ln\left(\frac{k_i}{2}r(t,\tau)\right) + \mathcal{A}_0(k_i r(t,\tau))\right]|y'(\tau)| + \mathcal{D}(x(t),y(\tau))|y'(\tau)|, \\
&= \frac{-|y'(\tau)|}{2\pi}J_0(k_i r(t,\tau))\left[\ln|t-\tau| + \ln\left(\frac{k_i}{2}\frac{r(t,\tau)}{|t-\tau|}\right)\right] \\
&\qquad + \frac{i}{4}\mathcal{A}_0(k_i r(t,\tau))|y'(\tau)| + \mathcal{D}(x(t),y(\tau))|y'(\tau)| \\
&= K_{5,1}(t,\tau)\ln|t-\tau| + K_{5,2}(t,\tau),
\end{aligned}$$

where

$$\begin{aligned}
K_{5,1}(t,\tau) &= \frac{-|y'(\tau)|}{2\pi}J_0(k_i r(t,\tau)), \\
K_{5,2}(t,\tau) &= K_{5,1}(t,\tau)\ln\left(\frac{k_i}{2}\frac{r(t,\tau)}{|t-\tau|}\right) + \frac{i}{4}\mathcal{A}_0(k_i r(t,\tau))|y'(\tau)| + \mathcal{D}(x(t),y(\tau))|y'(\tau)|.
\end{aligned}$$

Here, $\frac{r(t,\tau)}{|t-\tau|} = \sqrt{1+\left(\frac{f(t)-f(\tau)}{t-\tau}\right)^2}$ goes to $|x'(t)|$ as $\tau \to t$, and $\mathcal{D}(x,y)$ is a smooth function of its arguments. Moreover, $J_0(z)$ and $\mathcal{A}_0(z)$ are even and smooth functions of their arguments, and $sgn(t-\tau)r(t,\tau)$ is a smooth function of $t$ and $\tau$. Therefore both kernels $K_{5,1}(t,\tau)$ and $K_{5,2}(t,\tau)$ are smooth, and their diagonal terms are

$$K_{5,1}(t,t) = \frac{-|x'(t)|}{2\pi}, \tag{5.27a}$$

$$K_{5,2}(t,t) = \left\{\left[\frac{-1}{2\pi}\ln\left(\frac{k_i}{2}|x'(t)|\right) + \frac{i}{4} - \frac{1}{2\pi}\gamma\right] + \mathcal{D}(x(t),y(t))\right\}|x'(t)|. \tag{5.27b}$$

To compute kernel $K_6(t,\tau)$, note first that

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{n}_y}G^i(x,y) &= \frac{\partial}{\partial \mathbf{n}_y}\left[\frac{i}{4}H_0^{(1)}(k_i|x-y|) + \mathcal{D}(x,y)\right] \\
&= \frac{i}{4}\left(-H_1^{(1)}(k_i|x-y|)\right)k_i\frac{\partial|x-y|}{\partial \mathbf{n}_y} + \frac{\partial \mathcal{D}(x,y)}{\partial \mathbf{n}_y},
\end{aligned} \tag{5.28}$$

and that

$$\frac{\partial|x-y|}{\partial \mathbf{n}_y} = \mathbf{n}_y\cdot\nabla_y\sqrt{(x_1-y_1)^2+(x_2-y_2)^2} = \frac{\mathbf{n}_y\cdot(y_1-x_1,y_2-x_2)}{|x-y|} \tag{5.29}$$

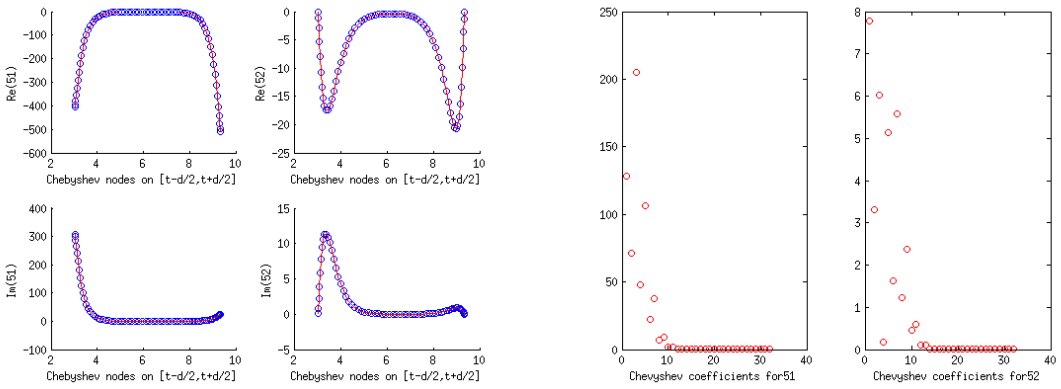



Figure 5.1: The Chebyshev interpolation of kernels $K_{5,1}$ and $K_{5,2}$ with 32 terms is accurate to 11 digits. Note the rapid decay of the Chebyshev coefficients for each kernel.

for all $y \neq x$. Using this with points $x(t) = (t, f(t))$ and $y(\tau) = (\tau, f(\tau))$ on the surface, and recalling that $\mathbf{n}_y = \frac{(-f'(\tau),1)}{|y'(\tau)|}$, the kernel $K_6(t,\tau)$ can be written as follows

$$\begin{aligned}
K_6(t,\tau) &= \frac{\partial G^i(x(t),y(\tau))}{\partial \mathbf{n}_y}|y'(\tau)| \\
&= \frac{-ik_i}{4}H_1^{(1)}(k_i r(t,\tau))\frac{(-f'(\tau),1)\cdot(\tau-t,f(\tau)-f(t))}{r(t,\tau)} + |y'(\tau)|\frac{\partial \mathcal{D}(x(t),y(\tau))}{\partial \mathbf{n}_y} \\
&= \frac{-ik_i^2}{4}c(\tau,t)\frac{H_1^{(1)}(k_i r(t,\tau))}{k_i r(t,\tau)} + |y'(\tau)|\frac{\partial \mathcal{D}(x(t),y(\tau))}{\partial \mathbf{n}_y},
\end{aligned}$$

where $c(\tau,t) = f(\tau) - f(t) - f'(\tau)(\tau - t)$. Using the series representation (A.8),

$$\begin{aligned}
K_6(t,\tau) &= \frac{-ik_i^2}{4}c(\tau,t)\left[\frac{-2iJ_0(k_i r(t,\tau))}{\pi[k_i r(t,\tau)]^2} + \frac{2iJ_1(k_i r(t,\tau))}{\pi k_i r(t,\tau)}\ln\left(\frac{k_i}{2}r(t,\tau)\right) + \mathcal{A}_2(k_i r(t,\tau))\right] \\
&\quad + |y'(\tau)|\frac{\partial \mathcal{D}(x(t),y(\tau))}{\partial \mathbf{n}_y} \\
&= -c(\tau,t)\frac{J_0(k_i r(t,\tau))}{2\pi r(t,\tau)^2} + \frac{k_i^2}{2\pi}c(\tau,t)\frac{J_1(k_i r(t,\tau))}{k_i r(t,\tau)}\left[\ln|t-\tau| + \ln\left(\frac{k_i}{2}\frac{r(t,\tau)}{|t-\tau|}\right)\right] \\
&\quad - \frac{ik_i^2}{4}c(\tau,t)\mathcal{A}_2(k_i r(t,\tau)) + |y'(\tau)|\frac{\partial \mathcal{D}(x(t),y(\tau))}{\partial \mathbf{n}_y} \\
&= K_{6,1}(t,\tau)\ln|t-\tau| + K_{6,2}(t,\tau),
\end{aligned}$$

where

$$K_{6,1}(t,\tau) = \frac{k_i^2}{2\pi} c(\tau,t) \frac{J_1(k_i r(t,\tau))}{k_i r(t,\tau)}, \tag{5.30a}$$

$$K_{6,2}(t,\tau) = K_{6,1}(t,\tau) \ln\left(\frac{k_i}{2}\frac{r(t,\tau)}{|t-\tau|}\right) - c(\tau,t)\left[\frac{J_0(k_i r(t,\tau))}{2\pi r(t,\tau)^2} + \frac{ik_i^2}{4}\mathcal{A}_2(k_i r(t,\tau))\right] + |y'(\tau)| \frac{\partial \mathcal{D}(x(t),y(\tau))}{\partial \mathbf{n}_y}. \tag{5.30b}$$

As can be seen from the series expansion (A.4b), $J_1(z)/z \to 1/2$ as $z \to 0$ and is a smooth and even function of its argument. Since $sgn(t-\tau)r(t,\tau)$ is smooth, the kernel $K_{6,1}(t,\tau)$ must be smooth as well. Similarly, $\mathcal{A}_2(z)$ is a smooth and even function (see (A.8)), so the $\mathcal{A}_2(k_i r(t,\tau))$ term is smooth. It remains then to show that the $\frac{c(\tau,t)}{2\pi r(t,\tau)^2} J_0(k_i r(t,\tau))$ term is smooth to conclude that the kernel $K_{6,2}(t,\tau)$ is also smooth. This can be seen by taking the limit as $\tau \to t$ and using l'Hospital's rule:

$$\lim_{\tau\to t} \frac{c(\tau,t)}{r(t,\tau)^2} = \lim_{\tau\to t} \frac{f(\tau) - f(t) - f'(\tau)(\tau - t)}{(t-\tau)^2 + (f(t) - f(\tau))^2} \tag{5.31}$$

$$= \lim_{\tau\to t} \frac{-f''(\tau)(\tau - t)}{-2(t-\tau) - 2f'(\tau)(f(t) - f(\tau))} \tag{5.32}$$

$$= \lim_{\tau\to t} \frac{-f'''(\tau)(\tau - t) - f''(\tau)}{2 - 2f''(\tau)(f(t) - f(\tau)) + 2f'(\tau)^2} \tag{5.33}$$

$$= \frac{-f''(t)}{2(1 + f'(t)^2)} \tag{5.34}$$

Therefore both kernels $K_{6,1}(t,\tau)$ and $K_{6,2}(t,\tau)$ are smooth, and their diagonal terms are

$$K_{6,1}(t,t) = 0, \tag{5.35a}$$

$$K_{6,2}(t,t) = \frac{f''(t)}{4\pi(1 + f'(t)^2)} + |x'(t)| \frac{\partial \mathcal{D}(x(t),y(t))}{\partial \mathbf{n}_y}. \tag{5.35b}$$

The kernel $K_7(t,\tau)$ is very similar to $K_6(t,\tau)$, and uses the normal derivative at $x$

$$\frac{\partial |x-y|}{\partial \mathbf{n}_x} = \frac{\mathbf{n}_x \cdot (x_1 - y_1, x_2 - y_2)}{|x-y|} \tag{5.36}$$

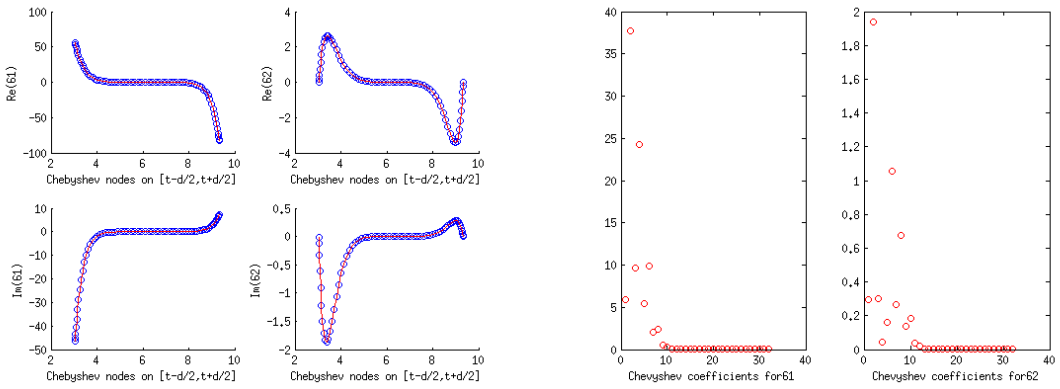



Figure 5.2: The Chebyshev interpolation of kernels $K_{6,1}$ and $K_{6,2}$ with 32 terms is accurate to 11 digits. Note the rapid decay of the Chebyshev coefficients for each kernel.

for $y \neq x$. Thus the decomposition is almost identical to that of $K_6(t,\tau)$:

$$\begin{aligned} K_7(t,\tau) &= \frac{\partial G^i(x(t),y(\tau))}{\partial \mathbf{n}_x}|y'(\tau)| \\ &= \frac{-ik_i^2}{4}c(t,\tau)\frac{H_1^{(1)}(k_i r(t,\tau))}{k_i r(t,\tau)} + |y'(\tau)|\frac{\partial \mathcal{D}(x(t),y(\tau))}{\partial \mathbf{n}_x} \\ &= K_{7,1}(t,\tau)\ln|t-\tau| + K_{7,2}(t,\tau), \end{aligned}$$

where

$$K_{7,1}(t,\tau) = \frac{k_i^2}{2\pi}c(t,\tau)\frac{J_1(k_i r(t,\tau))}{k_i r(t,\tau)}, \tag{5.37a}$$

$$\begin{aligned} K_{7,2}(t,\tau) = K_{7,1}(t,\tau)\ln\left(\frac{k_i}{2}\frac{r(t,\tau)}{|t-\tau|}\right) - c(t,\tau)\left[\frac{J_0(k_i r(t,\tau))}{2\pi r(t,\tau)^2} + \frac{ik_i^2}{4}\mathcal{A}_2(k_i r(t,\tau))\right] \\ + |y'(\tau)|\frac{\partial \mathcal{D}(x(t),y(\tau))}{\partial \mathbf{n}_x} \end{aligned} \tag{5.37b}$$

are smooth for the same reasons, noting that

$$\lim_{\tau\to t}\frac{c(t,\tau)}{r(t,\tau)^2} = \lim_{\tau\to t}\frac{c(\tau,t)}{r(t,\tau)^2} = \frac{-f''(t)}{2(1+f'(t)^2)}. \tag{5.38}$$

The diagonal terms in this case are

$$K_{7,1}(t,t) = 0, \tag{5.39a}$$

$$K_{7,2}(t,t) = \frac{f''(t)}{4\pi(1+f'(t)^2)} + |x'(t)|\frac{\partial \mathcal{D}(x(t),y(t))}{\partial \mathbf{n}_x}. \tag{5.39b}$$

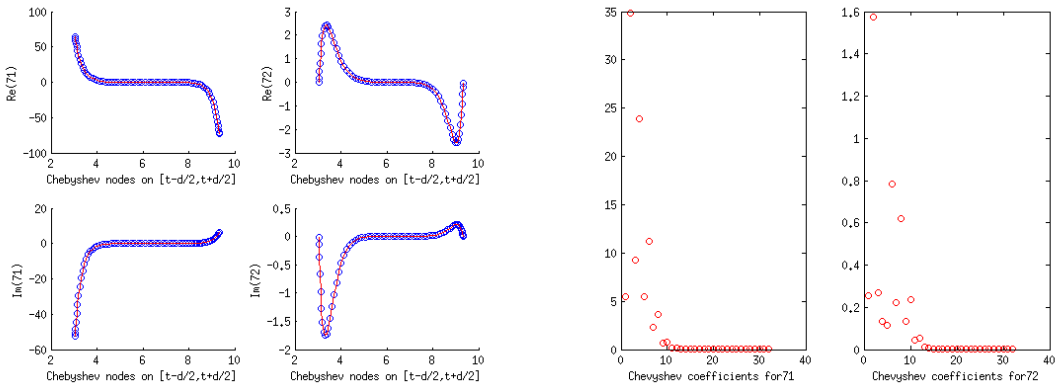



Figure 5.3: The Chebyshev interpolation of kernels $K_{7,1}$ and $K_{7,2}$ with 32 terms is accurate to 8 digits. Note the rapid decay of the Chebyshev coefficients for each kernel.

### 5.3.1 Numerical evaluation of weakly singular integrals $I_5$, $I_6$, $I_7$, $I_9$

Recall that

$$I_5(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_5(t,\tau)\xi_2(\tau)d\tau, \quad I_6(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_6(t,\tau)\xi_1(\tau)d\tau, \tag{5.40a}$$

$$I_7(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_7(t,\tau)\xi_2(\tau)d\tau, \quad I_9(t) = k_i^2\mathbf{n}(t)\cdot\int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_5(t,\tau)\mathbf{n}(\tau)\xi_1(\tau)d\tau, \tag{5.40b}$$

Since all the kernels here are logarithmic and similar to kernels $K_1$ through $K_4$, these integrals can be evaluated using previously described methods. As $\xi_1(t)$ and $\xi_2(t)$ are known quasi-periodic functions, they can be well approximated by truncated Floquet expansions:

$$\xi_1(\tau) \approx \sum_{n=-N}^{N} f_n^1 e^{i\alpha_n\tau}, \qquad \xi_2(\tau) \approx \sum_{n=-N}^{N} f_n^2 e^{i\alpha_n\tau}, \tag{5.41a}$$

$$n_1(\tau)\xi_1(\tau) \approx \sum_{n=-N}^{N} g_n^1 e^{i\alpha_n\tau}, \qquad n_2(\tau)\xi_1(\tau) \approx \sum_{n=-N}^{N} g_n^2 e^{i\alpha_n\tau}, \tag{5.41b}$$

where the coefficients can be computed efficiently via an FFT. Using these approximation in the integral expressions,

$$I_5(t) \approx \sum_{n=-N}^{N} f_n^2 W_5^n(t), \qquad I_6(t) \approx \sum_{n=-N}^{N} f_n^1 W_6^n(t), \tag{5.42a}$$

$$I_7(t) \approx \sum_{n=-N}^{N} f_n^2 W_7^n(t), \qquad I_9(t) \approx k_i^2 \sum_{n=-N}^{N} (n_1(t) g_n^1 + n_2(t) g_n^2) W_5^n(t), \tag{5.42b}$$

where

$$W_5^n(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_5(t,\tau) e^{i\alpha_n \tau} d\tau, \qquad W_6^n(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_6(t,\tau) e^{i\alpha_n \tau} d\tau, \tag{5.43}$$

$$W_7^n(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_7(t,\tau) e^{i\alpha_n \tau} d\tau. \tag{5.44}$$

Recall from the previous sections that kernels $K_5$, $K_6$ and $K_7$ have a logarithmic singularity.

## 5.4 Decomposition of the hyper-singular kernel $K_8$

To isolate the singularity of kernel $K_8$, recall that the quasi-periodic Green's function can be written as follows:

$$G^i(x,y) = \frac{i}{4} H_0^{(1)}(k_i|x-y|) + \mathcal{D}(x,y), \tag{5.45}$$

where $\mathcal{D}$ is an analytic function [52]. Moreover, $\left(H_0^{(1)}\right)' = -H_1^{(1)}$, thus the derivative of $G^i$ becomes:

$$\frac{d}{dt} G^i(x(t), y(\tau)) = \frac{d}{dt}\left(\frac{i}{4} H_0^{(1)}(k_i r(t,\tau)) + \mathcal{D}(x(t), y(\tau))\right) \tag{5.46}$$

$$= \frac{-ik_i}{4} H_1^{(1)}(k_i r(t,\tau)) \frac{dr(t,\tau)}{dt} + \nabla_x \mathcal{D}(x(t), y(\tau)) \cdot x'(t), \tag{5.47}$$

where $r(t,\tau) = |x(t) - y(\tau)| = \sqrt{(t-\tau)^2 + (f(t) - f(\tau))^2}$. The singular part of this expression comes from $H_1^{(1)}$, which can be expressed as

$$\begin{aligned} H_1^{(1)}(z) =& J_1(z) + iY_1(z) \\ =& J_1(z) - \frac{2i}{z\pi} + \frac{2i}{\pi}\left[\ln\left(\frac{z}{2}\right) + \gamma\right] J_1(z) - \frac{i}{\pi} \sum_{p=0}^{\infty} \frac{(-1)^p}{p!(p+1)!} \left(\frac{z}{2}\right)^{1+2p} (h_{p+1} + h_p) \\ =& \frac{-4}{ik_i}\left(\frac{-k_i}{2\pi z} + L(z)\right), \end{aligned}$$

where $L(z) = \frac{-ik_i}{4} H_1^{(1)}(z) + \frac{k_i}{2\pi z}$ is a smooth function of $z$, with $\lim_{z\to 0} L(z) = 0$ (see appendix A.1). Kernel $K_8$ can then be expressed as follows:

$$
\begin{aligned}
|x'(t)|K_8(t,\tau) =& \frac{d}{dt} G^i(x(t), y(\tau)) && (5.48)\\
=& \frac{-ik_i}{4} H_1^{(1)}(k_i r(t,\tau)) \frac{dr(t,\tau)}{dt} + \nabla_x \mathcal{D}(x(t), y(\tau)) \cdot x'(t) && (5.49)\\
=& \frac{dr(t,\tau)}{dt} \left[ \frac{-1}{2\pi r(t,\tau)} + L(k_i r(t,\tau)) \right] + \nabla_x \mathcal{D}(x(t), y(\tau)) \cdot x'(t) && (5.50)\\
=& \frac{-i}{d} \frac{1}{1 - e^{\frac{2\pi i}{d}(\tau - t)}} + \underbrace{\frac{dr(t,\tau)}{dt} \frac{-1}{2\pi r(t,\tau)} - \frac{-i}{d} \frac{1}{1 - e^{\frac{2\pi i}{d}(\tau - t)}}}_{M(t,\tau)} \\
& + \frac{dr(t,\tau)}{dt} L(k_i r(t,\tau)) + \nabla_x \mathcal{D}(x(t), y(\tau)) \cdot x'(t), && (5.51)
\end{aligned}
$$

where $M(t,\tau)$ is smooth (see appendix A.3) with $\lim_{\tau\to t} M(t,\tau) = \frac{i}{2d}\left(1 - \frac{d}{2\pi i} \frac{f''(t) f'(t)}{1 + f'(t)^2}\right)$, and $\frac{dr(t,\tau)}{dt}$ is bounded as $\tau \to t$ (see appendix A.2), so $\lim_{\tau\to t} \frac{dr(t,\tau)}{dt} L(k_i r(t,\tau)) = 0$. Thus the singularity in kernel $K_8(t,\tau)$ can be isolated as follows

$$
\begin{aligned}
K_8(t,\tau) =& \underbrace{\frac{-i}{d|x'(t)|} \frac{1}{1 - e^{\frac{2\pi i}{d}(\tau - t)}}}_{K_{8,1}(t,\tau)} \\
& + \underbrace{\frac{M(t,\tau)}{|x'(t)|} + \frac{L(k_i r(t,\tau))}{|x'(t)|} \frac{dr(t,\tau)}{dt} + \nabla_x \mathcal{D}(x(t), y(\tau)) \cdot \frac{x'(t)}{|x'(t)|}}_{K_{8,2}(t,\tau)} && (5.52)
\end{aligned}
$$

Here, the kernel $K_{8,2}$ is smooth, and the diagonal term is

$$
K_{8,2}(t,t) = \frac{i}{2d|x'(t)|} \left(1 - \frac{d}{2\pi i} \frac{f''(t) f'(t)}{1 + f'(t)^2}\right) + \nabla_x \mathcal{D}(x(t), y(t)) \cdot \frac{x'(t)}{|x'(t)|}. \qquad (5.53)
$$

## 5.5 Numerical evaluation of hypersingular singular integral $I_8$

Given the decomposition of kernel $K_8$ introduced in the previous section, the integral $I_8$ can be written as

$$
I_8(t) = \underbrace{\int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_{8,1}(t,\tau) \xi_1'(\tau) d\tau}_{I_{8,1}(t)} + \underbrace{\int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_{8,2}(t,\tau) \xi_1'(\tau) d\tau}_{I_{8,2}(t)}, \qquad (5.54)
$$

where $K_{8,1}$ is hypersingular and $K_{8,2}$ is smooth. To evaluate the hypersingular integral, $\xi_1'$ is expressed as a Floquet series:

$$\xi_1'(\tau) \approx \sum_{n=-N}^{N} h_n e^{i\alpha_n \tau}, \tag{5.55}$$

leading to

$$I_{8,1}(t) \approx \sum_{n=-N}^{N} h_n \underbrace{\int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_{8,1}(t,\tau) e^{i\alpha_n \tau} d\tau}_{W_{8,1}^n(t)}. \tag{5.56}$$

The hypersingular integral $W_{8,1}^n(t)$ is to be understood as a Cauchy principal value near the point $\tau = t$, and can be calculated explicitly as a complex integral over the smooth path $\gamma : [t - \frac{d}{2}, t + \frac{d}{2}] \to \{|z| = 1\}$ defined by $\gamma(\tau) = e^{\frac{2\pi i}{d}(\tau - t)}$:

$$W_{8,1}^n(t) = \frac{-i}{d|x'(t)|} \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} \frac{e^{i\alpha_n \tau}}{1 - e^{\frac{2\pi i}{d}(\tau - t)}} d\tau \tag{5.57}$$

$$= \frac{-1}{2\pi |x'(t)|} e^{i\alpha_n t} \int_{|z|=1} \frac{z^{(\alpha_n \frac{d}{2\pi} - 1)}}{1 - z} dz. \tag{5.58}$$

Note that the singularity at $z = 1$ is integrable as a Cauchy principal value, and the singularity at $z = 0$ is also integrable as long as the exponent $c = \alpha_n \frac{d}{2\pi} - 1 > -1$, as will be shown shortly. Recalling that $\alpha_n = \alpha + \frac{2\pi}{d} n$, the case $n > -\frac{\alpha d}{2\pi}$ guarantees that $c > -1$. The case $n \leq -\frac{\alpha d}{2\pi}$ can be handled by considering the complex conjugate of $W_{8,1}^n(t)$:

$$\overline{W_{8,1}^n(t)} = \frac{i}{d|x'(t)|} \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} \frac{e^{-i\alpha_n \tau} d\tau}{1 - e^{\frac{-2\pi i}{d}(\tau - t)}} \tag{5.59}$$

$$= \frac{-i}{d|x'(t)|} e^{\frac{-2\pi i}{d} t} \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} \frac{e^{-i\alpha_{n-1} \tau} d\tau}{1 - e^{\frac{2\pi i}{d}(\tau - t)}} \tag{5.60}$$

$$= \frac{-1}{2\pi |x'(t)|} e^{\frac{-2\pi i}{d} t} e^{-i\alpha_{n-1} t} \int_{|z|=1} \frac{z^{-\alpha_{n-1} \frac{d}{2\pi}}}{1 - z} \frac{dz}{z} \tag{5.61}$$

$$= \frac{-e^{-i\alpha_n t}}{2\pi |x'(t)|} \int_{|z|=1} \frac{z^{-\alpha_n \frac{d}{2\pi}}}{1 - z} dz, \tag{5.62}$$

where the integration path is the same as the one used for $W_{8,1}^n(t)$. The exponent of $z$ is now $c = -\alpha_n \frac{d}{2\pi} = -\frac{\alpha d}{2\pi} - n \geq 0$, since $n \leq -\frac{\alpha d}{2\pi}$. Thus it is sufficient to evaluate

$$I(c) = \int_{|z|=1} \frac{z^c}{1 - z} dz \tag{5.63}$$

for the case $c > -1$, which is done in appendix A.4. The value of the integral is then given by the expression

$$W^n_{8,1}(t) = \frac{-e^{i\alpha_n t}}{2\pi|x'(t)|} \begin{cases} -i\pi + i\sin\left(\frac{\alpha_n d}{2}\right)\left[\Psi\left(\frac{1}{2}+\frac{\alpha_n d}{4}\right) - \Psi\left(\frac{\alpha_n d}{4}\right)\right], & \text{if } n > \frac{-\alpha d}{2\pi} \\ i\pi - i\sin\left(\frac{\alpha_n d}{2}\right)\left[\Psi\left(1-\frac{\alpha_n d}{4}\right) - \Psi\left(\frac{1}{2}-\frac{\alpha_n d}{4}\right)\right], & \text{if } n \leq \frac{-\alpha d}{2\pi}, \end{cases}$$

where $\Psi$ is the digamma function.

## 5.6 Numerical integration with smooth kernel $K_{82}$

The numerical evaluation of the smooth integral

$$I_{8,2}(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_{8,2}(t,\tau)\xi_1'(\tau)d\tau \tag{5.64}$$

can be done either directly or by introducing a Floquet approximation:

$$\xi_1'(\tau) \approx \sum_{n=-N}^{N} h_n e^{i\alpha_n \tau}, \tag{5.65}$$

as was done for kernels $K_{5,2}$, $K_{6,2}$ or $K_{7,2}$. While it is tempting to do this, here this approach does not work so well, as will be shown below. Rather, a direct quadrature is shown to perform better.

### Floquet approximation of $\xi_1'$

Introducing the Floquet approximation of $\xi_1'(t)$ in the integral $I_{8,2}$,

$$I_{8,2}(t) \approx \sum_{n=-N}^{N} h_n \underbrace{\int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_{8,2}(t,\tau)e^{i\alpha_n\tau}d\tau}_{W^n_{8,2}(t)}. \tag{5.66}$$

The weights $W^n_{8,2}$ can then be approximated by using a Chebyshev interpolant for $K_{8,2}$ and the pre-computed values for Chebyshev polynomials integrated against $e^{in\pi x}$.

This approach does not yield similar accuracy as before however, despite kernel $K_{8,2}$ being smooth, just like kernels $K_{5,2}$, $K_{6,2}$ and $K_{7,2}$. Indeed, the kernel has a strong variation near the center of the integration interval, leading the Chebyshev interpolation to converge more slowly, as shown in figures 5.4 and 5.5. The interpolation is only

accurate to 3 digits even when 128 interpolation points are used. In contrast, the Chebyshev interpolations of kernels $K_{5,1}$, $K_{5,2}$, $K_{6,1}$, $K_{6,2}$, $K_{7,1}$ and $K_{7,2}$ are accurate to at least 8 digits using only 32 interpolation points. Here the accuracy was estimated as the mean squared difference between the kernel and its interpolant at twice as many Chebyshev nodes as were used to fit that interpolant.

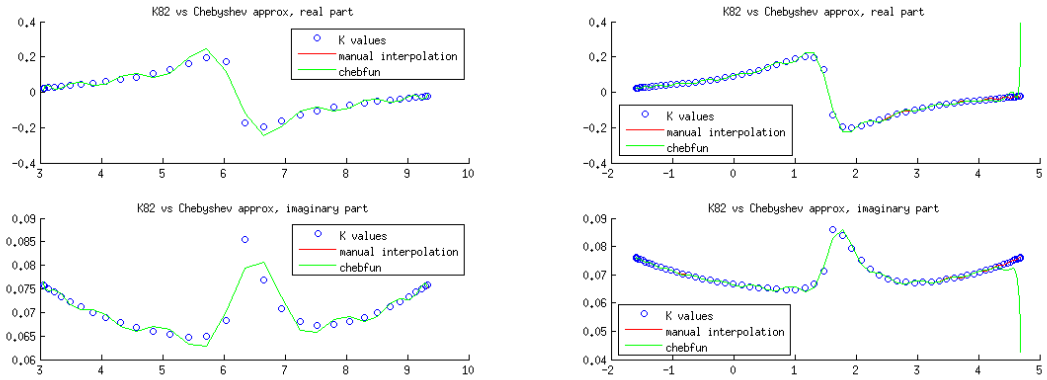



Figure 5.4: The Chebyshev interpolation of kernel $K_{8,2}$ with 16 and 32 terms is accurate only to 2 digits. The interpolations from the Chebfun package are identical to machine precision for 16 terms, but starts to differ from the manual interpolation with 32 terms.

The weights $W^n_{8,2}(t)$ may also be evaluated using other quadratures. The trapezoidal rule, as well as Matlab's buil-in adaptive quadratures quad and quadgk were tested. Quadgk was used since it may most efficient for high accuracy and oscillatory integrands, as is the case here, and it can handle moderate singularities at the endpoints (Matlab help files).

For the trapezoidal rule, the number of nodes is specified. For quad, a tolerance level is specified, and the number of function evaluations needed to reach that tolerance is output by quad. In the case of quadgk, the maximum number of function evaluations is specified, and quadgk returns an error bound when it reaches either the maximum number of evaluations or machine precision. The resulting convergence of the integral $I_{8,2}$ as a function of number of function evaluation is shown in the two figures below, where the measured error is the difference in the integral value from one iteration to the next.

Note that both Chebyshev interpolation and trapezoidal rule perform similarly in

this case, with only about 9 digits of accuracy for 4096 kernel evaluations. Matlab's built-in quadratures both perform worse, with quad stalling if asked for a tolerance of less than $10^{-7}$, and quadgk losing accuracy if allowed to use more than 256 nodes. This could be due to the oscillatory nature of the integrand, $K_{8,2}(t,\tau)e^{i\alpha_n\tau}$, or to the fact that $K_{8,2}(t,\tau)$ is hard to evaluate numerically for $t \approx \tau$, even though it is a smooth function at $t = \tau$.

**Direct quadrature of $K_{8,2}(t,\tau)\xi_1'(\tau)$**

Since the oscillatory integrand introduced by the Floquet approximation is completely artificial, and seems to cause problems with the numerical integration, another approach is to bypass the Floquet approximation and use a direct quadrature for

$$I_{8,2}(t) = \int_{t-\frac{d}{2}}^{t+\frac{d}{2}} K_{8,2}(t,\tau)\xi_1'(\tau)d\tau. \tag{5.67}$$

This may be done since $K_{8,2}(t,\tau)$ is known and can be easily evaluated at most points $\tau$, except perhaps $\tau \approx t$. The same is true of $\xi_1'(\tau)$.

Using the trapezoidal rule, the convergence (figure 5.8) is no different than the trapezoidal rule used in conjunction with the Floquet approximation, with 4097 function evaluations achieving less than 9 digits of accuracy. The only advantage being that now the quadrature must be performed only once, as opposed to $2N+1$ times for the Floquet approximation.

Using a Chebyshev approximation with the same number of points, the accuracy increases slightly to just over 9 digits. Looking at the distribution of the interpolation error measured at 8192 Chebyshev nodes, it is apparent that the integrand is not sufficiently well approximated by the interpolant near the center of the interval ($\tau \approx t$). One reason for this could be that while the interpolant itself has a high variation near $\tau = t$, the Chebyshev nodes used for the interpolation are concentrated away from this point. Ideally, to take full advantage of a Chebyshev quadrature, the integrand should have the largest variation near the endpoints. However, this cannot be achieved here by a shift of the integrand, since the integrand at $\tau + d/2$ is in no way related to the integrand at $\tau$. Instead, to better resolve the large variation near $\tau = t$, the integration interval can be split in two: $\tau < t$ and $\tau > t$, and a Chebyshev quadrature can be used on each interval.

Using this split Chebyshev quadrature, the integration error does not improve much, though the interpolation error seems to be even more concentrated around the point $\tau = t$, even though this point is now the end-point of two integration intervals, and as such has a high concentration of Chebyshev nodes. This seems to indicate an adaptive quadrature would be more suitable.

Using Matlab's built-in adaptive quadrature function quad, the error does not improve past 9 digits of accuracy, for a comparable number of function evaluations. Quadgk diverges after 256 function evaluations.

### Direct quadrature using linear interpolation of $K_{8,2}(t,\tau)$ near $\tau = t$

Since none of the quadrature tested above achieve an integration error below $10^{-9}$, despite using a smooth, non-oscillatory integrand and adaptive quadratures, it could be the case that the evaluation of $K_{8,2}(t,\tau)$ itself near $\tau = t$ is causing the loss of accuracy.

Note that while $K_{8,2}(t,\tau)$ itself is smooth, the expression in (5.52) is not easily evaluated. Instead, $K_{8,2}$ is evaluated numerically as $K_8 - K_{8,1}$. Since both $K_8(t,\tau)$ and $K_{8,1}$ are singular near $\tau = t$, their difference has a large round-off error, as can be seen in figure 5.12, where $K_{8,2}(t,\tau)$ is evaluated at 1000 points in the interval $t + [-10^{-3}, 10^{-3}]$.

A first-order solution to this problem is to linearly interpolate $K_{8,2}$ on this interval using the values $K_{8,2}(t, t - 10^{-5})$ and $K_{8,2}(t, t + 10^{-5})$. Thus, for $|\tau - t| < 10^{-5}$,

$$K_{8,2}(t,\tau) \approx K_{8,2}(t,\tau^-) + (K_{8,2}(t,\tau^+) - K_{8,2}(t,\tau^-))\frac{\tau - \tau^-}{\tau^+ - \tau^-}, \tag{5.68}$$

$$\tau^{\pm} = t \pm 10^{-5}. \tag{5.69}$$

Neither the trapezoidal rule nor the Chebyshev quadrature with up to $2^{12}$ function evaluations are affected by this, since neither evaluate points close enough to $\tau = t$.

For the split Chebyshev interpolation, up to 30 points need to be evaluated with the linear approximation, which improves the Chebyshev interpolation by 4 digits near $\tau = t$. However the integration error only improves marginally.

The biggest improvement occurs when the adaptive quadratures are used, with quad now achieving 12 digits of accuracy with around 1200 function evaluations, and quadgk achieving the same accuracy with only 32 function evaluations.

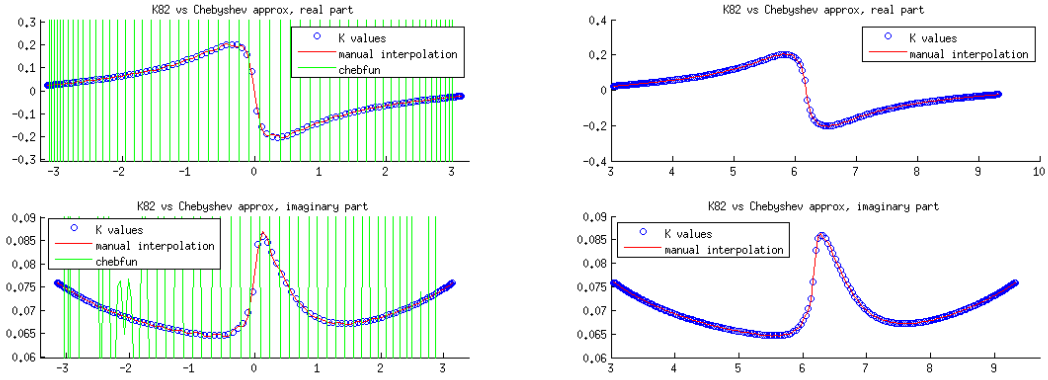



Figure 5.5: With 64 terms, the Chebyshev interpolation of kernel $K_{8,2}$ is accurate only to 3 digits. The Chebfun overfits badly. With 128 terms, the accuracy is still around 3 digits.

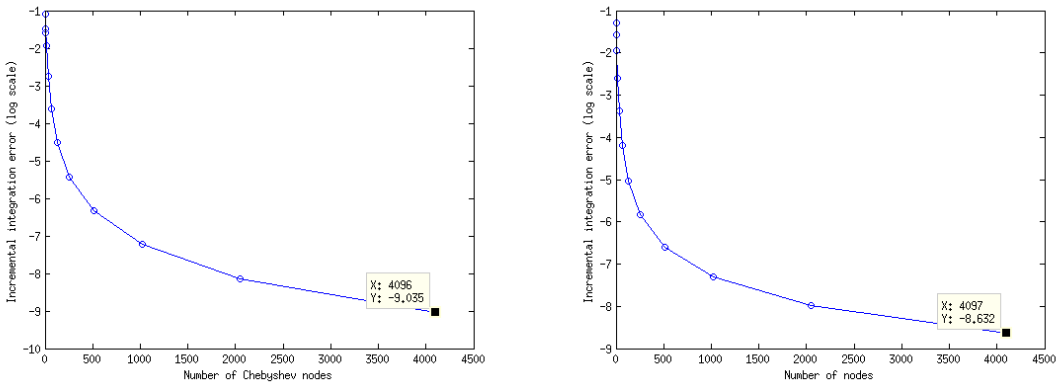



Figure 5.6: Convergence error for integral $I_{8,2}$ using Floquet approximation of $\xi_1'$ and Chebyshev interpolation of kernel $K_{8,2}$ (right), versus trapezoidal rule directly on integral $W_{8,2}^n$ (left).

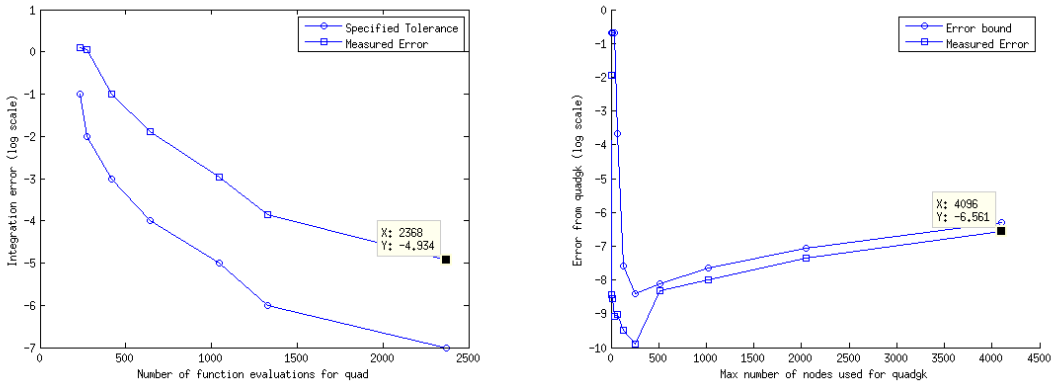



Figure 5.7: Convergence error for integral $I_{8,2}$ using Floquet approximation of $\xi_1'$ and Matlab's built-in adaptive quadrature quad (right), versus quadgk (left).

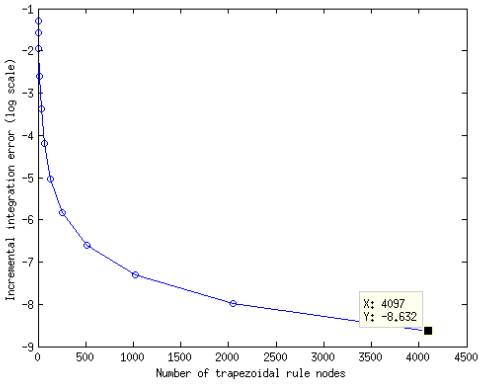



Figure 5.8: Convergence error for integral $I_{8,2}$ using the trapezoidal rule.

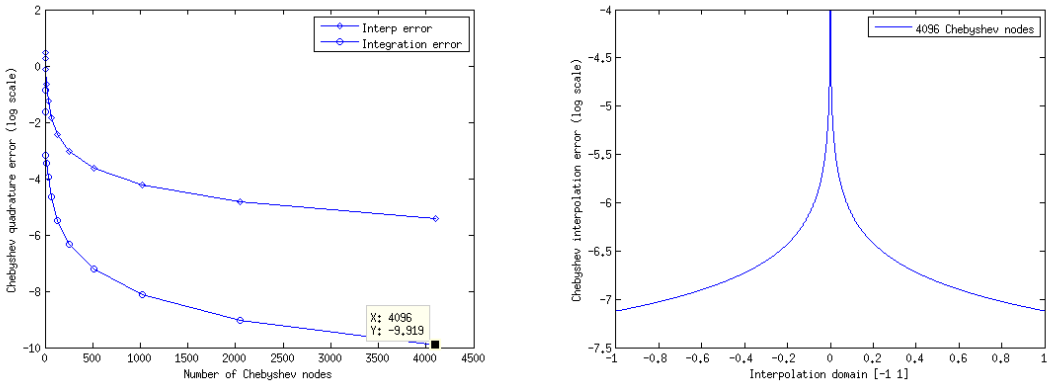



Figure 5.9: Convergence error for integral $I_{8,2}$ using a Chebyshev interpolation of the kernel $K_{8,2}(t,\tau)\xi_1'(\tau)$ (left), along with the interpolation error distribution on the interval $[-1,1]$ (right).

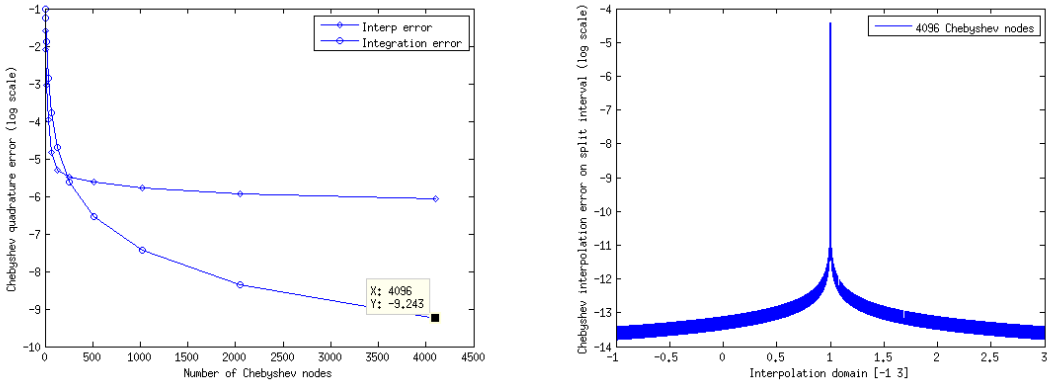



Figure 5.10: Convergence error for integral $I_{8,2}$ using a split Chebyshev interpolation of the kernel $K_{8,2}(t,\tau)\xi_1'(\tau)$ (left), along with the interpolation error distribution on both integration intervals (right).

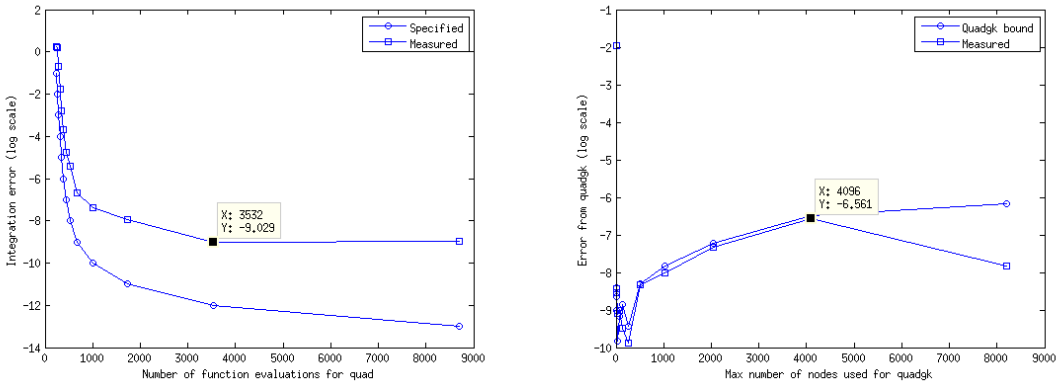



Figure 5.11: Convergence error for integral $I_{8,2}$ using Matlab's built-in adaptive quadratures quad (left) and quadgk (right).

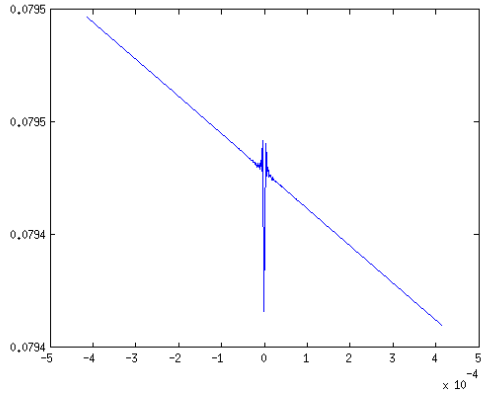



Figure 5.12: Numerical noise in the evaluation of the smooth kernel $K_{8,2}$.

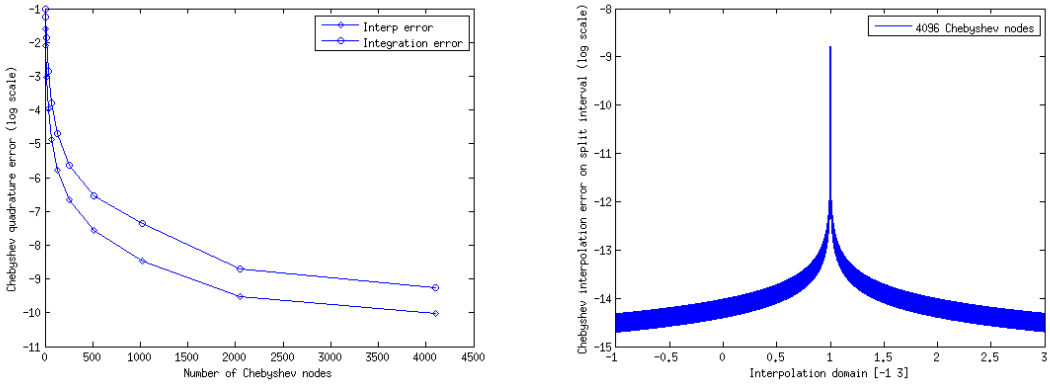



Figure 5.13: Convergence error for integral $I_{8,2}$ using a split Chebyshev interpolation of the kernel $K_{8,2}(t,\tau)\xi_1'(\tau)$ (left), along with the interpolation error distribution on both integration intervals (right).

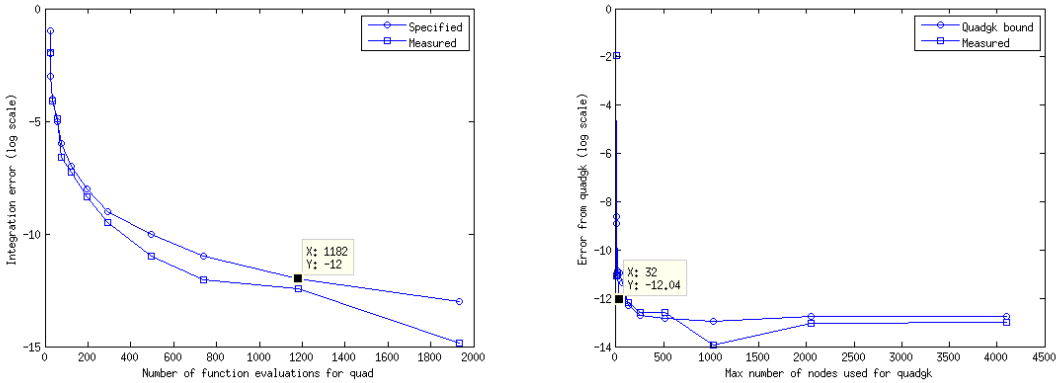



Figure 5.14: Convergence error for integral $I_{8,2}$ using Matlab's built-in adaptive quadratures quad (left) and quadgk (right).

# Chapter 6

# Derivative-Free Optimization

The choice of an optimization method depends on the objective function that will be optimized. Different choices of objective function shall be discussed in detail in the next chapter, but for now it is sufficient to observe that, for the purposes of designing nanoplasmonic gratings, the objective function will always depend on the electromagnetic field resulting from the illumination of the grating by some incident wave. While this field can be obtained using the integral equations solver outlined previously, the forward solver is a highly non-linear function of the grating profile, with no guarantees of convexity that would greatly help with the optimization. Thus a robust optimization method must be chosen.

Consider the following optimization problem: given a grating whose profile is described by a function $f(t, \mathbf{p})$, where $\mathbf{p} = (p_1, p_2, ..., p_r)$ are parameters controlling the geometry of the profile, and the resulting field $u(t, f(t, \mathbf{p}))$, minimize a given objective function $J(u)$ over all parameters $\mathbf{p}$ in a given parameter space $\Omega \in \mathbb{R}^r$.

Optimizing the objective function would greatly benefit from information contained in the derivatives of $J(p)$. In the absence of accurate gradient information, this could be approximated by finite-differences or other methods, but doing so is computationally expensive, as it requires at least 2 function evaluations per parameter to approximate a single gradient with second order accuracy. While the forward solver is relatively fast (as compared to previously used methods in nanoplasmonics), repeating this at every iteration of the optimization algorithm may still be too expensive in a virtual design setting. In light of this observation, derivative-free methods will be used here.

## 6.1 Derivative-Free Trust Region Algorithm

Recalling that the objective function of interest is non-linear, a second-order optimization algorithm is selected, as it can capture curvature information. The algorithm used here searches for a local minimum in the objective function by approximating it with a quadratic interpolation model, then minimizing or at least sufficiently decreasing the latter within a trust region. If the new point also decreases the objective function sufficiently, then it is accepted as a new iterate. This "sufficient" decrease, which shall be made precise below, guarantees the global convergence of the algorithm to second-order critical points.

For a point to be a minimum of a second-order model $m$, it is sufficient to be a stationary point where the Hessian is positive definite. Thus finding a minimum of the model can be achieved by minimizing both the size of the gradient and the size of the most negative eigenvalue of the Hessian (if it exists), both captured in the following measure of second-order stationarity [64]:

$$\sigma = \max\{\|\nabla m\|, -\lambda_{min}(H(m))\}.$$

The algorithm can be divided into six broad steps listed below [65]. The initialization step is only done once, while the other five are repeated until some stopping criteria is met.

**0. Initialization Step.** Decide on the initial point, trust region and model.

**1. Criticality Step.** If the stationarity of the model at the current iterate is small, meaning that the algorithm is close to a stationary point, make sure the model is more accurate by reducing trust-region radius and improving model.

**2. Step Calculation.** Use current model as a proxy for finding a trial point that may reduce the objective function on the trust region.

**3. Acceptance of Trial point.** Check that the trial point indeed reduces the objective function sufficiently.

**4. Model Improvement.** Update the model to include the trial point and check it is fully quadratic on the possibly updated trust region.

**5. Trust-Region Radius Update.** Update the radius of the trust-region depending on the reduction in objective function in step 3.

The stopping criteria may be reaching a maximum number of iterations $k_{max}$, having the second-order stationarity $\sigma_k^m$ within some tolerance of 0, or having the value of the objective function itself $J(p_k)$ within some tolerance of 0, when that makes sense. The steps of the algorithm are now laid out in detail.

**0. Initialization Step.** Decide on the initial point, trust region and model.

1. Fix the maximum trust region radius $\Delta_{max}$.
2. Choose an initial point $p_0$ and trust region radius $\Delta_0 \in (0, \Delta_{max})$.
3. Choose an initial interpolation set $Y_0$ with the correct dimension for the search space. A minimal positive basis with uniform angles is chosen here as the initial interpolation set.
4. Construct an initial fully-quadratic interpolation model $m_0$ on the trust region $B(p_0, \Delta_0)$, and compute its gradient $g_0$, its Hessian $H_0$ at $p_0$, as well as the second-order stationarity

$$\sigma_0^m = \max\{\|g_0\|, -\lambda_{min}(H_0)\}.$$

5. Choose thresholds $\eta_0$ and $\eta_1$ such that $0 \leq \eta_0 \leq \eta_1 < 1$ and $0 < \eta_1$ that will determine whether the relative reduction of the objective function is good, acceptable or unacceptable.
6. Choose trust-region radius increase and decrease factors $\gamma_{inc}$ and $\gamma_{dec}$ such that

$$0 < \gamma_{dec} < 1 < \gamma_{inc}.$$

7. Choose criticality threshold $\epsilon_c > 0$ below which the second-order stationarity of the model must be on the same scale as the trust-region radius.
8. Choose bounds $\mu > \beta > 0$ for the ratio between the trust-region radius and the stationarity of the model, in the case in which the latter is below the criticality threshold.
9. Choose $\omega \in (0, 1)$ as the incremental decrease factor for the trust-region radius in the criticality step.

10. Choose multiplicative factor $r$ to allow points up to $r\Delta_k$ away from the current iterate to be kept at the acceptance of trial-point step.
11. Set $k = 0$.

**1. Criticality Step.** Check that the model is fully-quadratic and that the trust-region radius is comparable with the measure of stationarity if the latter is close to 0.

**If** $\sigma_k^m > \epsilon_c$ keep current model and trust-region radius.

**Otherwise** check whether the model is already fully-quadratic or not.

**If the model is certified as being fully quadratic and** $\Delta_k \leq \mu\sigma_k^m$ keep current model and trust-region.

**Otherwise** gradually decrease trust region radius until it's comparable to the stationarity. Set $i = 0$. Set $m_k^{(0)} = m_k$ and $\Delta_k^{(0)}$.

1. While $\Delta_k^{(i)} > \mu\sigma_k^{m^{(i)}}$:
   (a) Decrease trust-region radius $\Delta_k^{(i+1)} = \omega\Delta_k^{(i)}$.
   (b) Call model-improvement algorithm on the model $m_k^{(i+1)}$ until it is fully-quadratic on $B(p_k, \Delta_k^{(i+1)})$.
   (c) Compute new stationarity $\sigma_k^{m^{(i+1)}}$ for model $m_k^{(i+1)}$.
   (d) Increment index $i$ by 1.
2. Use the new, possibly changed model $m_k = m_k^{(i)}$.
3. Set the new trust-region radius $\Delta_k = \min\{\max\{\Delta_k^{(i)}, \beta\sigma_k^{m^{(i)}}\}, \Delta_k\}$.

**2. Step Calculation.** Use current model as a proxy for finding a trial point that may reduce the objective function on the trust region. Compute step $s_k$ such that $p_k + s_k \in B(p_k, \Delta_k)$ and that $m(p_k + s_k)$ sufficiently decreases the model $m$.

A decrease that is sufficient for convergence to second-order critical points is one that is bounded below by a fraction of the decrease achieved by minimizing in the direction of the steepest descent. Moreover, since we are dealing with a quadratic model, the decrease must also be similarly related to the decrease achieved by minimizing in the direction of the greatest negative curvature.

These two conditions are quantified by finding the Cauchy step and the eigenstep respectively. The find the Cauchy step, the model $m$ is minimized in the direction of the

steepest descent $-\|g_k\|$, while remaining inside the trust region $B(p_k, \Delta_k)$:

$$t^C = \underset{\substack{t>0 \\ p_k - tg_k \in B(p_k, \Delta_k)}}{\operatorname{argmin}} m(p_k - tg_k).$$

Then the Cauchy step is given by $s_k^C = -t^C g_k$. The eigenstep $s^E$ minimizes the model in the direction of the most negative eigenvalue of the Hessian $H_k$ of the model, assuming it has at least one negative eigenvalue. Thus $s^E$ satisfies

$$H_k s^E = \lambda_{min}(H_k) s^E.$$

To ensure that $m(p_k + s^E)$ does indeed provide a decrease in the model, the direction of $s^E$ is chosen such that $s^E \cdot g_k < 0$. The negative curvature guarantees that the minimum along this direction is achieved when $p_k + s^E$ is on the boundary of the trust region, therefore the magnitude of the eigenstep must be $\|s^E\| = \Delta_k$. The best decrease provided by either the Cauchy step or the eigenstep is called the optimal decrease, and is given by $m(p_k) - \min\{m(p_k + s^C), m(p_k + s^E)\}$.

Thus a step $s_k$ is said to sufficiently decrease the model $m$ at $p_k$ if it decreases it by more than a constant fraction $\kappa_{fod} \in (0, 1]$ of the optimal decrease:

$$m(p_k + s_k) \geq \kappa_{fod}[m(p_k) - \min\{m(p_k + s^C), m(p_k + s^E)\}].$$

In practice, such a step can be calculated by computing the Cauchy step and possibly the eigenstep if $H_k$ has negative eigenvalues, and choosing the one that provides the best decrease in the model.

**3. Step Acceptance of Trial point.** Check that the trial point indeed reduces the objective function sufficiently. The previous step guarantees that the decrease in the model $m(p_k) - m(p_k + s_k)$ is sufficiently large. This step ensures that the decrease in the objective function itself is comparable, by computing the fraction

$$\rho_k = \frac{f(p_k) - f(p_k + s_k)}{m(p_k) - m(p_k + s_k)}.$$

Depending on whether the model $m_k$ can be certified as fully-quadratic and on the value $\rho_k$ and the thresholds $\eta_0$ and $\eta_1$ ($0 \leq \eta_0 \leq \eta_1 < 1$, $\eta_1 \neq 0$) fixed in the initialization step, four cases are distinguished:

**Successful:** If $\rho_k \geq \eta_1$, accept the new point $p_k + s_k$ as the new iterate $p_{k+1}$.

**Acceptable:** If $\eta_1 > \rho_k \geq \eta_0$ and the model $m_k$ is fully quadratic, accept the new point $p_{k+1} = p_k + s_k$ as the current iterate, but reduce the trust-region radius in step 5.

**Model improving:** If $\eta_1 > \rho_k$ and the model $m_k$ is not fully-quadratic, reject the new iterate and improve the model in the next step.

**Unsuccessful:** If $\rho_k < \eta_0$ and the model $m_k$ is fully-quadratic, reject the new iterate and decrease the trust-region radius in step 5.

In all of these cases (whether the iterate changes or not), the trial point $p_k + s_k$ is added to the sample set $Y_k$. Even if this point is not the new iterate, its inclusion may improve the model, and the costly function evaluation has already been done. Hence the new point is simply added to the set $Y_k$. The center of the trust region is moved to the new iterate $p_{k+1}$, and points outside the ball of radius $r\Delta_k$ around $p_{k+1}$ are discarded from the set $Y_k$. Since the set may no longer be suitable for a fully-quadratic interpolation (it may have too many or too few points for example), the model-improvement algorithm is applied to obtain a new model $m_{k+1}$.

**4. Model Improvement.** If $\rho_k < \eta_1$, call model-improvement algorithm until model is fully quadratic, and use this as the model $m_{k+1}$ for the next iteration.

**5. Trust-Region Radius Update.** The radius of the trust-region is updated depending on the case determined at the trial acceptance step:

**Successful,** i.e. $\rho_k \geq \eta_1$: the trust-region radius is increased by $\gamma_{inc}$ up to the maximum allowed, $\Delta_{k+1} = \min\{\gamma_{inc}\Delta_k, \Delta_{max}\}$.

**Acceptable or unsuccessful,** i.e. $\rho_k < \eta_1$ and the model $m_k$ was found to be fully quadratic: the trust-region radius is decreased by a fixed factor $\gamma_{dec}$, $\Delta_{k+1} = \gamma_{dec}\Delta_k$.

**Model improving,** i.e. $\rho_k < \eta_1$ and the model $m_k$ was not fully quadratic: the trust-region radius remains unchanged, $\Delta_{k+1} = \Delta_k$.

The iterate $k$ is incremented by 1 and steps one through five are repeated until the stopping criteria is met.

## 6.2 Implementation

The implementation of the DFO algorithm relies heavily on certain parameter choices, as well as the choice of a starting point and starting model. Moreover, the standard algorithm description leaves certain coding details open to interpretation. This sections attempts to cover all such choices made in this particular implementation of the DFO algorithm.

### 6.2.1 Choice of parameters

The performance of the DFO algorithm is highly sensitive to the choice of a dozen optimization parameters fixed throughout the algorithm.

Probably the most critical parameters are the choice of a starting point and trust region radius for the initial model. A good choice of starting point relies heavily on some prior knowledge of the features of the objective function. Generally, when dealing with a nanoplasmonic structure, domain knowledge usually defines a range of configuration parameters where the minimum is expected to occur. Thus choosing starting parameters in the middle of their ranges when known, and choosing a 0 starting point for those parameters with no known range. A complementary approach is to use multiple random starting point (using known ranges where possible) and see which leads to the largest decrease in objective function after a set number of iterations.

Experience shows that the maximum trust region radius must be on the same scale as the size of the largest feature of the objective function, while the initial trust region should be large enough to capture both the initial guess and the true minimum. Since these parameters effectively act as upper bounds to the step size of the algorithm, choosing $\Delta$'s that are too small may slow down the algorithm, while choosing $\Delta$'s that are too large may lead the the algorithm to miss out on smaller features of the function in the beginning, before it has a chance to appropriately reduce the trust region radius. Given these observations, $\Delta_{max}$ is generally take to be on the same scale as either the largest known feature of the objective function, or the radius of the search region. Given $\Delta_{max}$, the initial trust region radius is taken as half of that value: $\Delta_0 = \frac{1}{2}\Delta_{max}$.

### 6.2.2 Choice of model

Once an initial point is selected, a model of the objective function must be built in a region of radius $\Delta_0$ around it. The initial model differs from subsequent models in that it does not have any previously evaluated points to rely on. Thus a set of $(n+1)(n+2)/2$ points must then be chosen to interpolate a quadratic model in $n$ dimensions. The points are chosen so as to ensure the set is $\Lambda$-poised, for the poisedness constant *Lambda* fixed in the algorithm. In order to do so, a minimal positive basis with uniform angles is used. The points lying $\Delta_0$ away along the $n+1$ basis vectors form a set that is poised for linear interpolation, as pointed out in [65]. This set, together with the initial point, is completed to a $\Lambda$-poised set using the model improvement algorithm.

## 6.3 Validation

In order to test the implementation of the DFO algorithm, a few standard optimization test functions with known global minima is used.

First and foremost, the algorithm is tested on the sphere function in $n$ dimensions:

$$f_{sphere}(\mathbf{x}) = \sum_{i=1}^{n} x_i^2,$$

for $n = 2$, $n = 3$ and $n = 9$, with a search range of $-5 \leq x_i \leq 5$. The true minimum being at $x_i = 0 \forall i$, a random initial point in the specified range is chosen. Since the algorithm uses a quadratic model to interpolate the underlying objective functions, it is expected to find the true minimum within a single iteration. This is indeed observed for all three values of $n$. The parameters used for these results, as well as the starting point for all the following optimization problems are:

$$\begin{array}{llll} \Delta_{max} = 20 & \Lambda = 10 & & \\ \epsilon_c = 10^{-2} & \mu = 1 & \beta = 4.5 & \omega = 1/2 \\ \eta_0 = 0 & \eta_1 = 0.5 & & \\ \gamma_{dec} = 0.2 & \gamma_{inc} = 2 & r = 1 & tol = 10^{-15}. \end{array}$$

In each of the three tests, the algorithm reduces the objective function to machine precision (i.e. below the specified tolerance of $10^{-15}$ within 11, 18, and 101 function

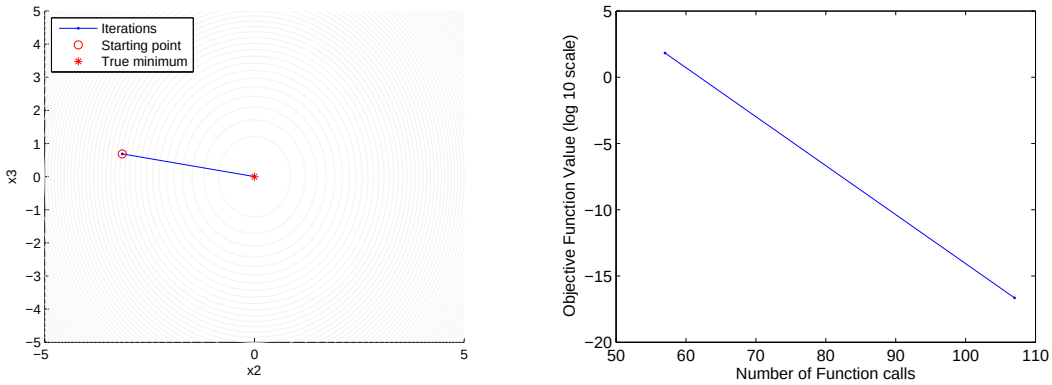



Figure 6.1: Convergence of the DFO algorithm on a quadratic test function in nine dimensions. The plot on the left shows the iterates on a two-dimensional cross-section of the contour plot of the function, while the plot on the right shows that machine precision is reached within a single iteration, requiring a total of 101 function calls.

calls respectively for the sphere functions in 2, 3 and 9 dimensions. Figure 6.1 illustrates the search path taken by the algorithm on a two-dimensional cross-section of the nine-dimensional sphere function. It also shows the error going from an initial error around 2 to below $10^{-15}$ in a single iteration. This outcome is observed across many different starting points in the given search range.

A slightly different test function is the Booth function in two dimensions:

$$f_{Booth}(\mathbf{x}) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2,$$

a skewed quadratic function with a 0 minimum at $\mathbf{x} = (1, 3)$. Using the same optimization parameters as before, convergence to the true minimum is achieved with 13 function calls, although as shown in figure 6.2, two iterations are needed here due to the shape of the objective function. Note that while another iteration was needed over the two-dimensional sphere function, only two extra function calls were made. This is because many of the points where the objective function is computed are reused from one iteration to another.

To test how the algorithm performs on functions with a nearly flat profile near the minimum, the Goldstein-Price function is used. The function is given by the following

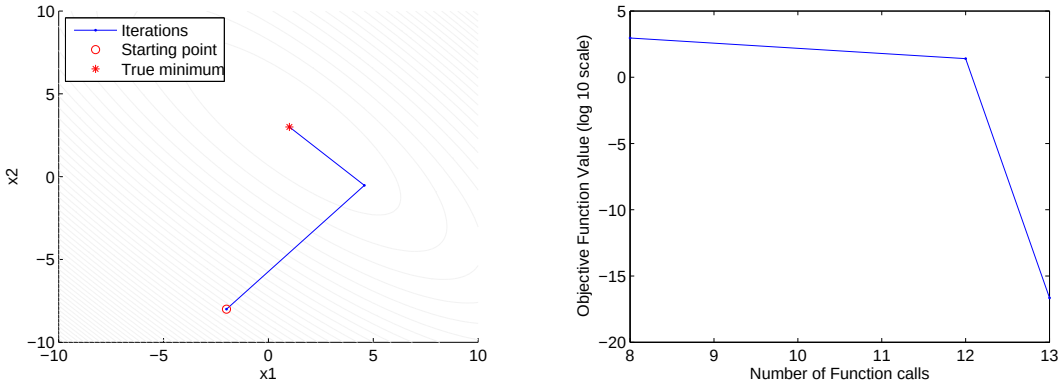



Figure 6.2: Convergence of the DFO algorithm on a quadratic test function in two dimensions. The plot on the left shows the iterates on a two-dimensional cross-section of the contour plot of the function, while the plot on the right shows that machine precision is reached within two iteration, requiring a total of 13 function calls.

equation:

$$f_{Goldstein-Price}(\mathbf{x}) = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) \\ (30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)) - 3,$$

and has its minimum at $f(0, -1) = 0$, its salient features appearing in the square $-2 \leq x_i \leq 2$. The starting point is placed outside the flat region at $(1.3, 1.8)$. The same parameters as before are used. The search predictably takes many more iterations than for the quadratic functions, as shown in figure 6.3, but converges steadily to the true minimum, which it reaches within machine precision after 127 function calls.

The next test function used is the Beale, a function similar to the Goldstein-Price, but with a much flatter profile, thus posing a greater challenge to the optimization algorithm. The profile

$$f_{Beale}(\mathbf{x}) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$$

is mostly flat in the square $-4.5 \leq x_i \leq 4.5$, with sharp peaks at the four corners and has a 0 minimum at $(3, 0.5)$. To really test out the algorithm, the initial guess is placed on one of the peaks, at $(-3, -4)$. Since the main feature of the function is larger than either of the functions considered before, the maximum trust region radius is increased. The

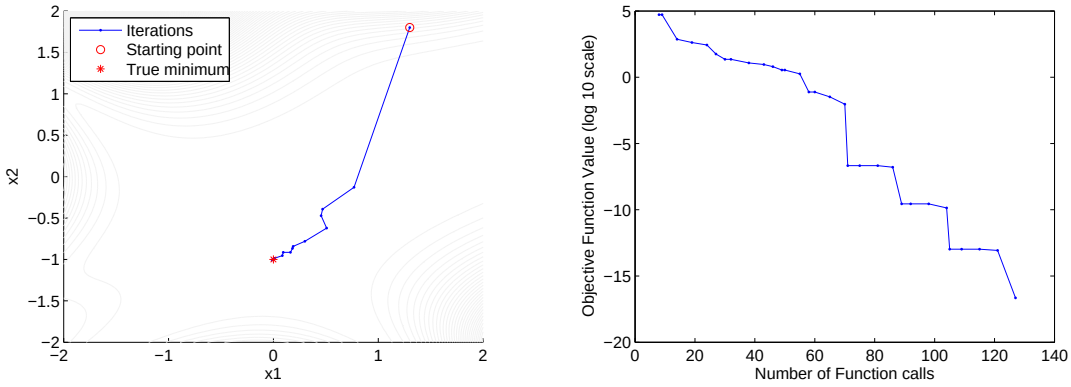



Figure 6.3: Convergence of the DFO algorithm on the Golstein-Price test function in two dimensions. The plot on the left shows the iterates on a two-dimensional cross-section of the contour plot of the function, while the plot on the right shows that machine precision is reached within two iteration, requiring a total of 127 function calls.

difficult flat profile also suggests the thresholds for acceptable and successful iteration be both increased. Thus the following parameters are changed from those used previously:

$$\Delta_{max} = 100 \qquad \eta_0 = 0.1 \qquad \eta_1 = 0.6.$$

Figure 6.4 shows the search path of the algorithm as it approaches the true minimum, which it reaches after 279 function evaluations.

A different kind of challenge to an optimization algorithm is posed by the Rosenbrock function, whose unique minimum lies in a steep and curved valley:

$$f_{Rosenbrock}(\mathbf{x}) = 100\left(x_2 - x_1^2\right)^2 + (x_1 - 1)^2.$$

The unique minimum lies at $(1, 1)$. Starting with an initial guess at $(-1, -1)$ forces the algorithm to follow the steep curved valley in order to converge to the correct point. Since the valley is relatively narrow, the maximum trust-region radius is reduced, while most other optimization parameters are kept at their default values:

$$\Delta_{max} = 0.1 \qquad \gamma_{dec} = 0.5 \qquad \gamma_{inc} = 2.5.$$

The convergence is shown in figure 6.5. As expected, after a relatively quick initial descent into the valley, convergence slows down as the algorithm follows the bottom

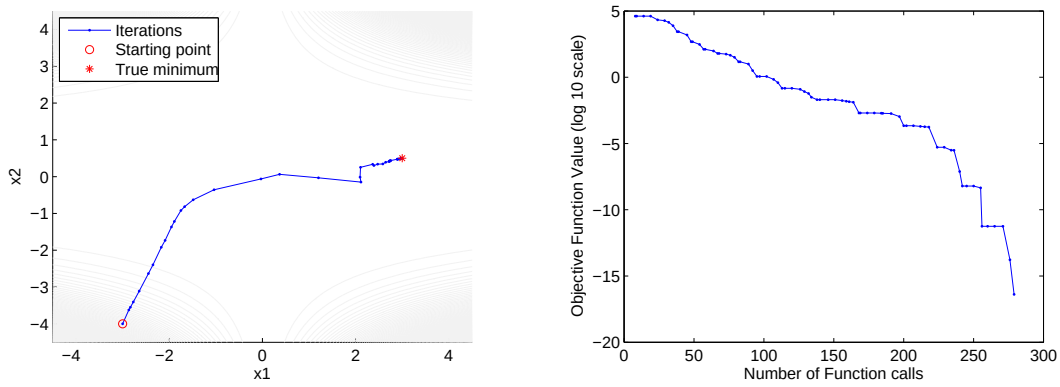



Figure 6.4: Convergence of the DFO algorithm on the Beale test function in two dimensions. The plot on the left shows the iterates on the contour plot of the function, which is mostly flat except for a sharp dip at the origin. while the plot on the right shows that machine precision is reached with 279 function calls.

of the valley, then speeds up again as it enters the nearly-quadratic region around the global minimum.

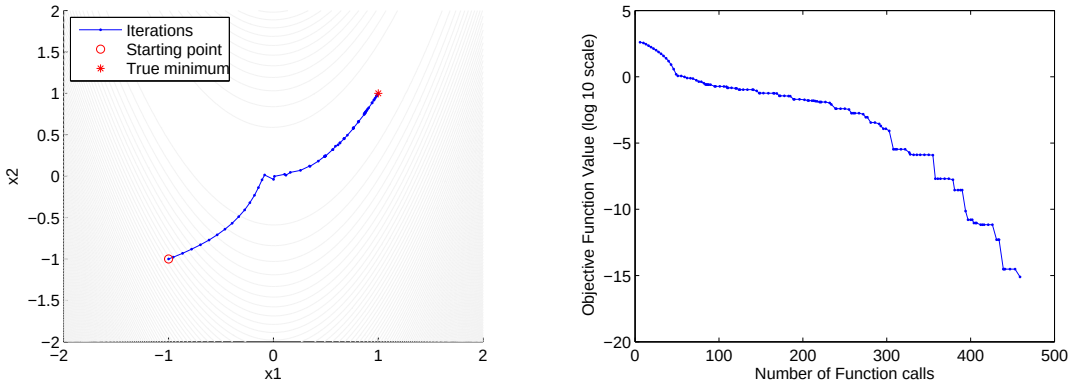



Figure 6.5: Convergence of the DFO algorithm on the Rosenbrock test function in two dimensions. The plot on the left shows the iterates on a contour plot of the function, while the plot on the right shows that machine precision is reached with 413 function calls.

# Chapter 7

# Optimal Design of Nanoplasmonic Surfaces

As discussed in Chapter 2, nanoplasmonic devices rely on the excitation of a surface plasmon polariton (SPP) on the interface between a dielectric and a conducting material, most often a metal. One way to excite these SPPs is to use a carefully designed, periodic corrugation of the metal surface [66]. It is the goal of this chapter to show how gratings that support SPPs can be designed using the integral equation methods and DFO algorithm described in previous chapters.

## 7.1 Objective functions

SPPs are electromagnetic waves that arise when an incident electromagnetic field excites the electron cloud at the surface of the metal at a critical resonant frequency. These waves propagate along the metal/dielectric interface and decay sharply away from it. Thus one way to design SPP-supporting grating surfaces is to directly maximize the surface field enhancement compared to the incident field.

For another approach, notice that for an SPP, the energy of the incident field is almost completely transferred into modes that decay in the far field, so that the reflected field is very small away from the interface. Finding a grating that sufficiently minimizes this reflectivity would also ensure that this grating supports SPPs.

Let's see what these objective functions translate to in terms of the quantities

introduced so far.

### 7.1.1 Maximum point-wise surface field

Consider, as in chapter 4, a metallic grating whose profile is described by the function $f(t)$, for $t \in [0, d]$. Consider also a monochromatic wave with wavelength $\lambda$, incident at an angle $\theta$ from the normal. This wave is described by

$$u^{inc}(x_1, x_2) = e^{i(\alpha x_1 - \beta x_2)},$$

where $\alpha = k_e \sin(\theta)$ and $\beta = k_e \cos(\theta)$, and $k_e = \frac{2\pi}{\lambda}$ denotes the exterior wave number.

An approximation of the resulting surface field is obtained by using the integral equation forward solver to solve for the Floquet coefficients $a_n$:

$$u(t) \approx \sum_{n=-N}^{n=N} a_n e^{i\alpha_n t}$$

where $\alpha_n = \alpha + 2n\pi/d$.

In order to maximize the point-wise enhancement of this field with respect to the incident field, the following objective function must be minimized:

$$J = min_{t\in[0,d]} \left\{ \frac{|u^{inc}(t)|}{|u(t)|} \right\}. \tag{7.1}$$

### 7.1.2 Minimal reflectivity

Instead of maximizing the surface field, one could also minimize the reflectivity in order to obtain a surface that supports SPPs. Recall from (4.17) the Rayleigh expansion of the scattered field,

$$u^s(x_1, x_2) = \sum_{n=-\infty}^{\infty} b_n^e e^{i(\alpha_n x_1 + \beta_n^e x_2)},$$

valid only for $x_2 > \max_{x_1}\{f(x_1)\}$, where $b_n^e$ are the exterior Rayleigh coefficients and $\beta_n^e = \sqrt{k_e^2 - \alpha_n^2}$. Note that only those modes $n$ for which $\beta_n^e > 0$ propagate away from the surface, while those where $\beta_n^e \in i\mathbb{R}$ decay rapidly away from the surface. The set of propagating modes is the set

$$U = \left\{ n : -|k_e| - \alpha < \frac{2n\pi}{d} < |k_e| - \alpha \right\},$$

and the reflectivity is the sum of the energies

$$e_n = \frac{\beta_n^e}{\beta}|b_n^e|^2$$

for each of the propagating nodes. As discussed before, a grating that supports SPPs is a grating where the propagating modes carry no energy away from the surface, or in other words the reflectivity

$$R = \sum_{n \in U} \frac{\beta_n^e}{\beta}|b_n^e|^2$$

must be much smaller than 1. This can be achieved by minimizing the objective function:

$$J = R. \tag{7.2}$$

## 7.2 Parameter search space

Note that each of the objective functions $J(u)$ described above depend on the grating profile $f(t)$ through the field $u(t, f(t))$. This profile $f(t, \mathbf{p})$, in turn, depends on a number of free parameters $\mathbf{p} = (p_1, p_2, ..., p_r)$ that can be varied over a certain parameter space $\Omega \in \mathbb{R}^r$ in order to minimize the objective functions described above. These parameters could be the period of the grating, its depth, as well as minimal curvature or maximal slope. More generally, the surface of a smooth two dimensional periodic grating considered in this work can be represented by a Fourier series, whose coefficients can be treated as free parameters. For simplicity, the search space is restricted to surfaces that can be represented by truncated Fourier cosine series:

$$f(t) = \sum_{k=1}^{r} p_k \cos\left(\frac{dkt}{2\pi}\right),$$

where as before, $d$ is the period of the grating, and where the vertical offset is assumed to be 0 since it does not affect any of the objective functions of interest here. The period $d$ itself could be used as a search parameter, though in practice this is the parameter that is least variable, since observation has shown that only gratings with periods close to the wavelength of the incident field support SPPs.

## 7.3 Implementation

The Derivative-Free Optimization algorithm described in Chapter 6 is used to minimize the objective functions described above. While the theory behind this algorithm is well-established, the implementation is tricky: the algorithm relies on the careful selection of more than a dozen different hyper-parameters, including a starting point, and there is no documented way of selecting them. Moreover, a model must also be selected for approximating the objective function within the trust-region. Based on experimentation using the standard test functions used to test the algorithm in the previous chapter, some of these parameters are more critical than others. The choices made for these critical parameters are outlined below. The other parameters values are left unchanged from those chosen in Chapter 6. Similarly, the same quadratic model, based on a $\Lambda$-poised set of points, and outlined in the previous chapter is used.

The first implementation parameter that must be fixed is the dimension $r$ of the parameter search space. Since this parameters determines the complexity of the grating's surface profile, it must be large enough that at least one gratings described with only $r$ Fourier coefficients can support SPPs. On the other hand, a large $r$ hampers the performance of the DFO algorithm. Through experimentation with different values, it is concluded that a value $r = 5$ results in both a reasonably tractable optimization problem, and very good final values of the objective function.

The second critical choice in this or any optimization problem is the choice of a starting point, or initial guess of a surface profile in this case. From the extensive literature on nanoplasmonic gratings emerges a pattern that the height of the gratings is always much smaller than their period, with heights typically no greater than 50nm for periods in the 400 - 700nm range. This suggests that 1) a flat profile would be a good starting point for the search space and 2) none of the 5 Fourier coefficients should be larger than 25nm, so the optimal solution should exist in a trust region of radius no larger than about 50nm. This conclusion informs the choice of the initial trust region radius $\Delta_0$ and the maximum trust region radius $\Delta_{max}$ by providing an upper bound. In practice, setting $\Delta_{max} = 50$nm is too loose of a constraint, since trust region centers can move at each iteration and end up far outside the ball of radius 50nm centered at 0. Instead, using $\Delta_{max} = 1$ and $\Delta_0 = 1/2$ has proven to achieve good objective function

values, and not get stuck in minima.

Once the search space, the initial profile and family of models are chosen, the optimization algorithm needs thresholds $\eta_0$ and $\eta_1$ for deciding between successful, acceptable and unsuccessful iterations, as detailed in Chapter 6. The higher these values, subject to $0 \leq \eta_0 \leq \eta_1 < 1$, the closer the quadratic model must be to the underlying objective function in order to be trusted. Since there is no reason to expect the objective function to be quadratic, it is expected that a quadratic model may not be a very good approximation. Therefore the threshold values for successful and acceptable iteration cannot be very stringent. The values $\eta_0 = 0$ and $\eta_1 = 0.5$ are used through most of the numerical experiments.

The last critical co ice for the path that the algorithm will take through the search space are the trust-region radius increase and decrease factors $\gamma_{inc}$ and $\gamma_{dec}$. Recall from Chapter 6 that these are the factors used to increase the radius of the current model's trust region in a successful iteration, or decrease it in the case of an acceptable or unsuccessful iteration. These largely determine the extent to which the algorithm is greedy vs cautious. In choosing these parameters, a balance must be struck between a greedy algorithm that can more easily be stuck in local minima and a very cautious algorithm that converges very slowly. After experimenting with different combinations of values, it seems that $\gamma_{inc} = 2$ and $\gamma_{dec} = 0.2$ work well for the application at hand.

## 7.4 Results

In this section different setups that could support SPPs are explored, and their exact geometry is optimized for the different objective functions described above. The exact implementation parameters vary in each case, as the complexity of the forward problem as well as the complexity of objective landscape itself vary. The values used in each case are found through a combination of grid-search and heuristic reasoning, as discussed below. All the different examples share the basic setup of a silver grating with period $d = 530$nm illuminated by an incoming plane wave with wavelength $\lambda = 555$nm. In all these examples, the profile is modeled as a Fourier series (as described in section 7.2) with 5 non-zero coefficients.

### 7.4.1 Minimal reflectivity under normal incidence

The following results arise from optimizing the grating profile when the incident field is incoming at 0 degrees from the vertical (normal). The objective here is to minimize the reflectivity of the surface. Since at normal incidence the forward problem is relatively simple, using $N = 64$ terms in the Floquet series approximations proves to be sufficient. The optimization landscape likewise turns out not to be very challenging. Once the maximum trust-region radius is reduced to $\Delta_{max} = 0.1$, using the relatively lax optimization threshold values $\eta_0 = 0$, $\eta_1 = 0.5$ and the relatively greedy trust-region increase and decrease parameters $\gamma_{dec} = 0.2$, $\gamma_{inc} = 2$ leads to excellent results.

Figure 7.1 shows the convergence of the DFO algorithm on the left, and the grating profile corresponding to the optimal point achieved on the right. Note that fewer than 600 evaluations (function calls) of the expensive objective function were necessary in order to bring the value of the reflectivity to machine precision. It is interesting to see that the height of the grating is around 40nm, confirming the initial assumption on the maximal height of SPP-supporting gratings.

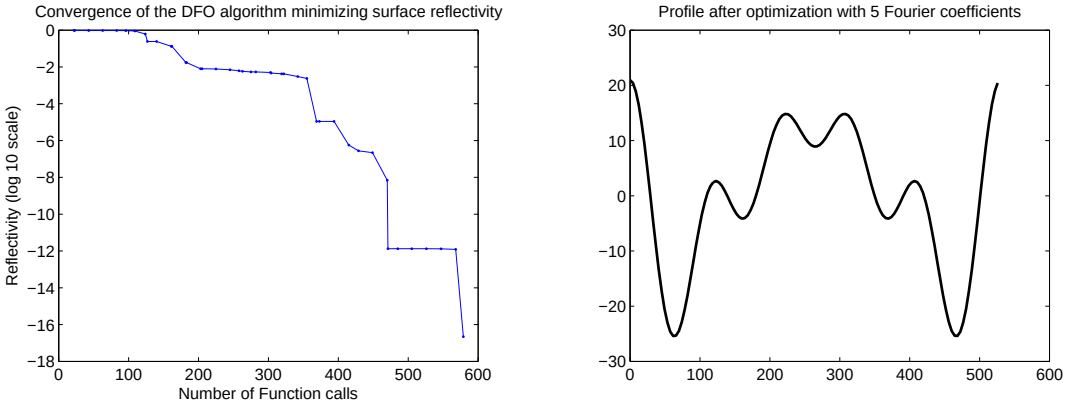



Figure 7.1: The DFO algorithm uses less than 600 evaluations of the objective function to reduce the reflectivity to machine precision 0. The corresponding grating profile, on the right, has a total height of around 40nm.

Figure 7.2 shows the magnitude of the electromagnetic field in a 1300nm band around the grating surface. It is clear from this image that the objective of the optimization was achieved and the final grating clearly exhibits a surface plasmon polariton. Almost no

energy is reflected far from the surface, while most of it is instead concentrated within the first 100nm above the surface, a scale much smaller than the 550nm wavelength. The field near the surface is also enhanced by a factor of at least 10, another clear indicator of an SPP.

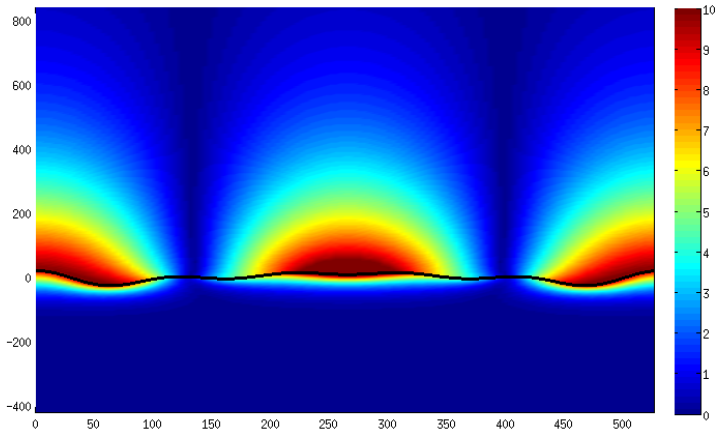



Figure 7.2: The field in a 1300nm band around the grating confirms the conclusion of the DFO algorithm: most of the energy is concentrated very close to the surface, where the field is highly enhanced.

### 7.4.2 Minimal reflectivity under oblique incidence

Using the same setup as before, the grating's reflectivity is optimized in the case when the incident plane wave is incoming at 27 degrees from the normal. In this case it is necessary to increase the accuracy of the forward solver, so a value of $N = 128$ is used for the number of Floquet series approximation terms. The optimization problem itself is also more challenging, requiring more stringent thresholds $\eta_0 = 0.25$ and $\eta_1 = 0.75$ for acceptable and successful iterations. The maximum trust-region radius and the increase/decrease parameters remain the same: $\Delta_{max} = 0.1$, $\gamma_{dec} = 0.2$, $\gamma_{inc} = 2$.

Figure 7.3 shows the convergence of the DFO algorithm on the left, and the grating profile corresponding to the optimal point achieved on the right. After 1000 evaluations (function calls) of the expensive objective function, the objective function achieves only a value of $10^{-2.5}$.

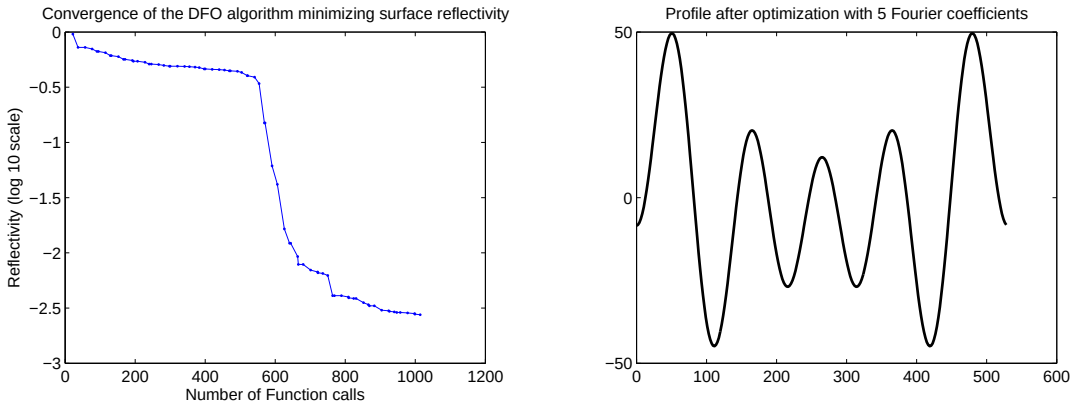



Figure 7.3: The DFO algorithm uses 1000 evaluations of the objective function to reduce the reflectivity to $10^{-2.5}$. The corresponding grating profile, on the right, has a total height of 100nm.

Figure 7.4 shows the magnitude of the electromagnetic field in a 1300nm band around the grating surface. It is clear from this image that the objective of the optimization was achieved and the final grating clearly exhibits a surface plasmon polariton. Almost no energy is reflected far from the surface, while most of it is instead concentrated within the first 100nm above the surface, a scale much smaller than the 550nm wavelength. The field near the surface is also enhanced by a factor of at least 7, another clear indicator of an SPP.

## 7.5 Discussion

This chapter has explored some ways in which the derivative-free optimization algorithm can be combined with the efficient integral equation solver to do computer-assisted design of nanoplasmonic gratings.

Firstly, the design problem has been inscribed within the framework of derivative-free optimization. To do so, the objective function has been clearly defined in terms of the output of the integral equation solver, and the optimization spaced has been defined as the space of truncated Fourier cosine functions used to model the grating profile. The choice of initial starting point for the optimization algorithm, as well as some fixed geometric parameters, are informed by domain knowledge about nanpoplasmonic

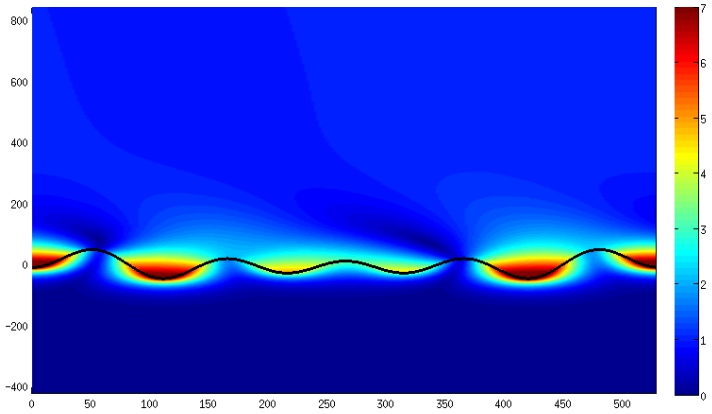



Figure 7.4: The field in a 1300nm band around the grating confirms the conclusion of the DFO algorithm: most of the energy is concentrated very close to the surface, where the field is highly enhanced.

surfaces.

Once the framework has been established, experimentation with the DFO algorithm hyperparameters leads to rapid convergence to minima of the objective functions. These minima, while not guaranteed to be global, solve the design problem in that the corresponding gratings do indeed support surface plasmons, as shown in figures 7.2 and 7.4.

In light of these results, the methods described herein show clear potential in the design of nanoplasmonic devices.

# Chapter 8

# Conclusion

The field of nanoplasmonics has enjoyed a rise in popularity in recent years. This popularity is deserved in light of its many success stories, ranging from near-field optical microscopes to nanoantennas to biosensors such as the home pregnancy test, but also the potential it holds for a wider range of biosensors, high-density optical storage or high-efficiency solar cells among others. Since nanodevices are highly sensitive to specific material properties and geometry in ways that are not yet completely understood, designing new devices relies heavily on trial-and-error. To speed up the process, numerical simulations are often used in lieu of trials, but there is still much room for improvement upon the state-of-the-art methods used in the field.

This thesis improves the design process in two ways: firstly by introducing a new simulation method based on integral equation methods that is much faster than the methods currently used in the field, and secondly by proposing the use of a derivative-free trust-region optimization algorithm to replace the manual trial-and-error design process itself. The overall achievement of this thesis is to show that these two methods can indeed be successfully used in the virtual design of nanoplasmonic surfaces.

The first important contribution of this thesis towards the final objective is the development of a new, efficient integral equation method for simulating an electromagnetic field near a periodic, 2D metallic nano-grating in a dielectric medium. This does not appear in the literature up to this point. As shown in chapter 4, the method relies on a frequency-domain formulation of the transmission problem, which leads to a system of Helmholtz equations that are in turn reformulated as a system of integral equations of

the second kind. Chapter 4 also details some of the challenges of this approach, such as the singularities in the integral equations kernels and the balance between precision and speed in the numerical evaluating the green's functions involved. These challenges are addressed by reformulating the integral equations in terms of a linear combination of Green's functions, and using three different Green's functions representations for different regions of the domain, depending on their convergence speed in that region.

The second important contribution of this thesis is the implementation of a derivative-free optimization algorithm and its application to the design of nanoplasmonic surfaces. While the DFO algorithm itself has been discussed in literature [65], implementation details are left to up to the reader. Moreover, the algorithm relies on over a dozen hyper-parameters, whose careful tuning has a large impact on performance. These implementation details and hyperparameter tuning strategies are discussed in chapter 6. Chapter 7 shows how the DFO algorithm can be used in conjunction with the integral equations solver to design nanoplasmonic surfaces. To achieve this, the design problem was recast in an optimization framework by defining an objective function and an optimization space. A few design examples are explored, and in each case the algorithm developed here is able to rapidly find solutions which correspond to metallic gratings that exhibit surface plasmons, as expected. This is a clear indication of the algorithm's potential applications in the design of nanoplasmonic devices.

There are a number of directions in which the current algorithm can be improved in order to make it more precisely suited to nanodevice manufacturers. Firstly, the optimization space could be expanded by either using additional Fourier terms in the surface representation, or using an entirely different geometric parametrization that more closely mimics the parameters that can be controlled in a fabrication setting. For an even larger space of feasible surfaces, the integral equations solver could to be extended to handle non-smooth interfaces. A further natural extension of the solver would be to 3D domains. This poses a number of challenges, such as finding a way to efficiently evaluate 3D Green's function for Maxwell's equations, and managing the increased computational complexity.

As far as the design problem is concerned, another promising research direction is the development of additional objective functions, better tailored to the real-life needs of nanodevice builders, who may be interested in for example a large reflectivity drop

over varying illumination wavelengths or angles.

In a related area, adapting the methods introduced here into an inverse problem solver is another exciting direction of future inquiry. The design problem solved here and the inverse problem naturally have fundamental differences, notably questions around uniqueness of solutions. But the algorithm developed in this thesis also has many of the elements of an inverse solver. Such an inverse solver could help with nanoplasmonic fabrication by providing a way to "see" the shape that is being built.

Finally, all the algorithms implemented thus far could benefit for an additional boost in computational performance by taking advantage of parallel computing architecture. Indeed this is a natural fit in many parts of the algorithm, as many of the Green's function or objective function evaluation, currently the performance bottlenecks, are fundamentally independent of each other.

# Bibliography


[1] Jord Cornelis Prangsma. *Local and dynamic properties of light interacting with subwavelength holes.* PhD thesis, University of Twente, 2009.

[2] Edward D. Palik. *Handbook of optical constants of solids II.* Academic Press, 1991.

[3] Mark I. Stockman. Nanoplasmonics: The physics behind the applications. *Physics Today*, 64(2):39–44, 2011.

[4] G. Mie. Contributions to the optics of turbid media, particularly of colloidal metal solutions. *25*, 3(Annalen der Physik):377–445, 1908.

[5] R. W. Wood. On a remarkable case of uneven distribution of light in a diffraction grating spectrum. *Proc. Phys. Soc. London*, 18(269), 1902.

[6] U. Fano. The theory of anomalous diffraction gratings and of quasi-stationary waves on metallic surfaces (sommerfeld's waves). *JOSA*, 31(3):213–222, 1941.

[7] Stefan A. Maier. *Plasmonics: Fundamentals and Applications.* Springer, 2007.

[8] T. W. Ebessen, H. J. Lezec, H. F. Ghaemi, T. Thio, and P. A. Wolff. Extraordinary optical transmission through sub-wavelength hole arrays. *Letters to Nature*, 391:667–669, 1998.

[9] E. Popov, M. Neviere, S. Enoch, and R. Reinisch. Theory of light transmission through subwavelength periodic hole arrays. *Physical Review B*, 62(23):16100–16108, 2000.

[10] F. J. Garcia-Vidal, L. Martin-Moreno, T. W. Ebbesen, and L. Kuipers. Light passing through subwavelength apertures. *Reviews of Modern Physics*, 82(1):729–787, 2010.

[11] Peter Zijlstra, James W. M. Chon, and Min Gu. Five-dimensional optical recording mediated by surface plasmons in gold nanorods. *Nature*, 459:410–413, 2009.

[12] Nathan C. Lindquist, Prashant Nagpal, Antoine Lesuffleur, David J. Norris, and Sang-Hyun Oh. Three-dimensional plasmonic nanofocusing. *Nano Letters*, 10(4):1369–1373, 2010.

[13] John David Jackson. *Classical Electrodynamics*. John Wiley & Sons, 1999.

[14] Tilo Arens and Thorsten Hohage. On radiation conditions for rough surface scattering problems. *IMA J Appl Math*, 70(6):839–847, 2005.

[15] Simon N. Chandler-Wilde and Bo Zhang. A uniqueness result for scattering by infinite rough surfaces. *SIAM J. APPL. MATH.*, 58(6):1774–1790, 1998.

[16] Fernando Reitich and Oscar Bruno. Solution of a boundary value problem for the helmholtz equation via variation of the boundary into the complex domain. *Proceedings of the Royal Society of Edinburgh*, 122A:317–340, 1992.

[17] W. Challener, I. Sendur, and C. Peng. Scattered field formulation of finite difference time domain for a focused light beam in dense media with lossy materials. *Opt. Express*, 11(23):3160–3170, 2003.

[18] Xianshi1 Lin and Xuguang Huang. Numerical modeling of a teeth-shaped nanoplasmonic waveguide filter. *Journal of the Optical Society of America B: Optical Physics*, 26(7):1263–1268, 2009.

[19] Shyamsunder Erramilli Ahmet A. Yanik, Ronen Adato and Hatice Altug. Plasmon hybridization in nanoapertures for development of an efficient nanoantenna array. *Proc. of SPIE*, 7394, 2009.

[20] R. Fikri, D. Barchiesi, F. HÕDhili, R. Bachelot, A. Vial, and P. Royer. Modeling recent experiments of apertureless near-field optical microscopy using 2d finite element method. *Optics Communications*, 221:13–22, 2003.

[21] David P. Lyvers, Jeong-Mi Moon, Alexander V. Kildishev, Vladimir M. Shalaev, and Alexander Wei. Gold nanorod arrays as plasmonic cavity resonators. *ACS Nano*, 2(12):2569–2576, 2008.

[22] Kai Stannigel, Michael König, Jens Niegemann, and Kurt Busch. Discontinuous galerkin time-domain computations of metallic nanostructures. *Opt. Express*, 17(17):14934–14947, 2009.

[23] J.S Hesthaven and T Warburton. Nodal high-order methods on unstructured grids: I. time-domain solution of maxwell's equations. *Journal of Computational Physics*, 181(1):186–221, 2002.

[24] Andreas M. Kern and Olivier J. F. Martin. Surface integral formulation for 3d simulations of plasmonic and high permittivity nanostructures. *J. Opt. Soc. Am. A*, 26(4):732–740, 2009.

[25] José M. Taboada, Javier Rivero, Fernando Obelleiro, Marta G Araújo, and Luis Landesa. Method-of-moments formulation for the analysis of plasmonic nano-optical antennas. *J. Opt. Soc. Am. A*, 28(7):1341–1348, 2011.

[26] Wei E. I. Sha, Wallace C. H. Choy, Yongpin P. Chen, and Weng Cho Chew. Optical design of organic solar cell with hybrid plasmonic system. *Optics Express*, 19(17):15908–15918, 2011.

[27] Babak Alavikia and Omar M. Ramahi. A hybrid finite element method - surface integral equation using quasi-periodic green's function in solving the problem of scattering from infinite periodic conducting grating. In *Antennas and Propagation Society International Symposium, IEEE*, 2010.

[28] K.S. Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *IEEE Trans. Antennas and Propagation*, 16:302–307, 1966.

[29] A. Taflove. *Advance in Computational Electro-dynamics: The Finite Difference Time Domain Method*. Artech House, 1998.

[30] Raju R. Namburu, Eric R. Mark, and Jerry A. Clarke. Scalable electromagnetic simulation environment. *CMES: Computer Modeling in Engineering & Sciences*, 5(4):443–454, 2004.

[31] Richard Holland. Pitfalls of staircase meshing. *IEEE Transactions on Electromagnetic Compatibility*, 35(4):434–439, 1993.

[32] S. Dey and R. Mittra. A locally conformal finite-difference time-domain (fdtd) algorithm for modeling three-dimensional perfectly conducting objects. *IEEE Micro. Guided Wave Lett.*, 7:273–275, 1997.

[33] Wenhua Yu, R. Mittra, Xiaoling Yang, Yongjun Liu, Qinjiang Rao, and A. Muto. High-performance conformal fdtd techniques. *IEEE Microwave Magazine*, 11(4):43–55, 2010.

[34] Agostino Monorchio and Raj Mittra. A hybrid finite-element finite-difference time-domain (fe/fdtd) technique for solving complex electromagnetic problems. *IEEE Microwave and Guided Wave Letters*, 8(2):93–95, 1998.

[35] T. Namiki. A new fdtd algorithm based on alternating-direction implicit method. *IEEE Trans. Microw. Theory Techn*, 47(10):2003–2007, 1999.

[36] F. Zheng, Z. Chen, and J. Zhang. A finite-difference time-domain method without the courant stability conditions. *IEEE Microwave and Guided Wave Letters*, 9(11):441–443, 1999.

[37] GERRIT MUR. Absorbing boundary conditions for the finite-difference approximation of the time-domain electromagnetic-field equations. *IEEE Trans. Electromagn. Compat.*, 23(4):377–382, 1981.

[38] Jean-Pierre Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185 – 200, 1994.

[39] BJORN ENGQUIST and ANDREW MAJDA. Absorbing boundary conditions for numerical simulation of waves. *Proc. Natl. Acad. Sci.*, 74(5):1765–1766, 1977.

[40] Peter Monk. *Finite Element Methods for Maxwell's Equations*. Clarendon Press, 2003.

[41] Bernardo Cockburn, Fengyan Li, and Chi-Wang Shu. Locally divergence-free discontinuous galerkin methods for the maxwell equations. *J. Comput. Phys.*, 194(2):588–610, 2004.

[42] Leonid Kucherov and Dan Givoli. High-order absorbing boundary conditions incorporated in a spectral element formulation. *International Journal for Numerical Methods in Biomedical Engineering*, 26(9):1130–1143, 2010.

[43] Seppo Matias Jarvenpaa. *Implementation of PML absorbing boundary condition for solving Maxwell's equations with Whitney elements*. PhD thesis, Helsingin Yliopisto (Finland), 2001.

[44] A. J. Poggio and E. K. Miller. Integral equation solutions of three-dimensional scattering problems. In Raj Mittra, editor, *Computer techniques for electromagnetics*, chapter 4. Oxford, New York, Pergamon Press, 1973.

[45] Rainer Kress. *Linear Integral Equations*. Springer, 1988.

[46] David Colton and Kress Rainer. *Inverse Acoustic and Electromagnetic Scattering Theory*. Springer-Verlag, 1992.

[47] Oscar Bruno. New high-order integral methods in computational electromagnetism. *CMES*, 5(4):319–330, 2004.

[48] R. E. Kleinman and G. F. Roach. Boundary integral equations for the three-dimensional helmholtz equation. *SIAM Review*, 16(2):214–236, 1974.

[49] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.

[50] N. Nishimura. Fast multipole accelerated boundary integral equation methods. *Appl Mech Rev*, 55(4):299–324, 2002.

[51] E. Bleszynski, M. Bleszynski, and T. Jaroszewicz. Aim: Adaptive integral method for solving large-scale electromagnetic scattering and radiation problems. *Radio Sci.*, 31(5):1225–1252, 1996.

[52] O. P. Bruno and M. C. Haslam. Efficient high-order evaluation of scattering by periodic surfaces: deep gratings, high frequencies, and glancing incidences. *J. Opt. Soc. Am. A*, 26(3):658–668, 2009.

[53] O. P. Bruno and M. C. Haslam. Efficient high-order evaluation of scattering by periodic surfaces: vector-parametric gratings and geometric singularities. *Waves in Random and Complex Media*, pages 1–21, 2010.

[54] Gunther Schmidt and Bernd H. Kleemann. Integral equation methods from grating theory to photonics: an overview and new approaches for conical diffraction. *Journal of Modern Optics*, 58(5-6):407–423, 2010.

[55] Ozgur Ergul and Levent Gurel. Comparison of integral-equation formulations for the fast and accurate solution of scattering problems involving dielectric objects with the multilevel fast multipole algorithm. *IEEE Transactions on antennas and propagation*, 57(1):176–187, 2009.

[56] Alexander Ihler. An overview of fast multipole methods. MIT Area Exam, 2004.

[57] Nail A. Gumerov and Ramani Duraiswami. Computation of scattering from clusters of spheres using the fast multipole method. *J. Acoust. Soc. Am.*, 117(4):1744–1761, 2004.

[58] Yoshihiro Otani and Naoshi Nishimura. A periodic FMM for Maxwell's equations in 3D and its applications to problems related to photonic crystals. *J. Comput. Phys.*, 227(9):4630–4652, 2008.

[59] Eric Darve and Pascal Havé. A fast multipole method for maxwell equations stable at all frequencies. *Phil. Trans. R. Soc. Lond. A*, 362:603–628, 2004.

[60] Li Hu, Le-Wei Li, and Raj Mittra. Electromagnetic scattering by finite periodic arrays using the characteristic basis function and adaptive integral methods. *IEEE Transactions of Antennas and Propagation*, 58(9):3086–3090, 2010.

[61] Oscar P. Bruno and Leonid A. Kunyansky. A fast, high-order algorithm for the solution of surface scattering problems: Basic implementation, tests, and applications. *JCP*, 169:80–110, 2001.

[62] David Colton and Kress Rainer. *Integral Equation Methods in Scattering Theory*. Wiley, 1983.

[63] Rainer Kress. On the numerical solution of a hypersingular integral equation in scattering theory. *Journal of Computational and Applied Mathematics*, pages 345–360, 1995.

[64] A. Conn, N. Gould, and P. Toint. *Trust Region Methods*. Society for Industrial and Applied Mathematics, 2000.

[65] A. Conn, K. Scheinberg, and L. Vicente. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, 2009.

[66] R. H. Ritchie, E. T. Arakawa, J. J. Cowan, and R. N. Hamm. Surface-plasmon resonance effect in grating diffraction. *Physical Review Letters*, 21(22):1530–1533, 1968.

[67] Milton Abramowitz and Irene Stegun. *Handbook of Mathematical Functions*. National Bureau of Standards, 1964.

# Appendix A

# Appendix

## A.1 Hankel functions

The Hankel functions of the first kind are defined as

$$H_\alpha^{(1)}(z) = J_\alpha(z) + iY_\alpha(z), \tag{A.1}$$

where $J_\alpha(z)$ and $Y_\alpha(z)$ are the Bessel functions of first and second kind respectively.

Their derivatives are given by

$$\frac{d}{dz}H_0^{(1)}(z) = -H_1^{(1)}(z), \tag{A.2}$$

$$\frac{d}{dz}H_1^{(1)}(z) = H_0^{(1)}(z) - \frac{1}{z}H_1^{(1)}(z). \tag{A.3}$$

Series representations for the Hankel functions can be obtained from the series representation of the Bessel function of the first kind ([67] 9.1.10),

$$J_0(z) = \sum_{p=0}^{\infty} \frac{(-1)^p}{(p!)^2}\left(\frac{z}{2}\right)^{2p}, \tag{A.4a}$$

$$J_1(z) = \sum_{p=0}^{\infty} \frac{(-1)^p}{p!(1+p)!}\left(\frac{z}{2}\right)^{1+2p}, \tag{A.4b}$$

and the series representation for the Bessel function of the second kind ([67] 9.1.11),

$$Y_0(z) = \frac{2}{\pi}\left[\ln\frac{z}{2} + \gamma\right]J_0(z) - \frac{2}{\pi}\sum_{p=0}^{\infty}\frac{(-1)^p}{(p!)^2}\left(\frac{z}{2}\right)^{2p}h_p, \tag{A.5a}$$

$$Y_1(z) = -\frac{2}{\pi z} + \frac{2}{\pi}\left[\ln\frac{z}{2} + \gamma\right]J_1(z) - \frac{1}{\pi}\sum_{p=0}^{\infty}\frac{(-1)^p}{p!(1+p)!}\left(\frac{z}{2}\right)^{1+2p}(h_{p+1} + h_p), \tag{A.5b}$$

where $h_p = \sum_{m=1}^{p} \frac{1}{m}$ is the $p^{th}$ harmonic number and $\gamma = \lim_{p\to\infty}[h_p - \ln p]$ is the Euler-Mascheroni constant. Note that $J_0(z)$ is an even function of $z$, while $J_1(z)$ is odd. More generally, $J_n(z)$ is an even function of $z$ if $n$ is even, and odd otherwise. Neumann's expansion of the Bessel function of the second kind ([67] 9.1.88) may also be useful:

$$Y_0(z) = \frac{2}{\pi}\left[\ln\frac{z}{2} + \gamma\right] J_0(z) - \frac{4}{\pi}\sum_{p=1}^{\infty}\frac{(-1)^p}{p} J_{2p}(z), \tag{A.6a}$$

$$Y_1(z) = -\frac{2}{\pi z} J_0(z) + \frac{2}{\pi}\left[\ln\frac{z}{2} + \gamma - 1\right] J_1(z) - \frac{2}{\pi}\sum_{p=1}^{\infty}\frac{(-1)^p(2p+1)}{p(p+1)} J_{2p+1}(z). \tag{A.6b}$$

Thus the following expansions are obtained for the Hankel's functions:

$$H_0^{(1)}(z) = \frac{2i}{\pi}\ln\frac{z}{2} J_0(z) + \underbrace{\left(1 + \frac{2i}{\pi}\gamma\right) J_0(z) - \frac{4i}{\pi}\sum_{p=1}^{\infty}\frac{(-1)^p}{p} J_{2p}(z)}_{\mathcal{A}_0(z)}, \tag{A.7a}$$

$$H_1^{(1)}(z) = -\frac{2i}{\pi z} J_0(z) + \underbrace{\left[1 + \frac{2i}{\pi}\left(\ln\frac{z}{2} + \gamma - 1\right)\right] J_1(z) - \frac{2i}{\pi}\sum_{p=1}^{\infty}\frac{(-1)^p(2p+1)}{p(p+1)} J_{2p+1}(z)}_{\mathcal{A}_1(z)}, \tag{A.7b}$$

where smoothness of the functions $\mathcal{A}_0$ and $\mathcal{A}_1$ follows from the smoothness of $J_n(z)$ for any integer $n$ and the smoothness of $J_n(z)/z$ for any $n \geq 1$. Note also that $\mathcal{A}_0(z)$ is an even function of $z$.

A series expansion for $H_1^{(1)}/z$ is also used in this thesis and can be obtained from the previous series

$$\begin{aligned}\frac{1}{z}H_1^{(1)}(z) = &-\frac{2i}{\pi z}\left[\frac{J_0(z)}{z} - \ln\frac{z}{2} J_1(z)\right] \\ &+ \underbrace{\left[1 + \frac{2i}{\pi}(\gamma - 1)\right]\frac{J_1(z)}{z} - \frac{2i}{\pi}\sum_{p=1}^{\infty}\frac{(-1)^p(2p+1)}{p(p+1)}\frac{J_{2p+1}(z)}{z}}_{\mathcal{A}_2(z)},\end{aligned} \tag{A.8}$$

where $\mathcal{A}_2(z)$ is a smooth, even function of $z$. The functions $\mathcal{A}(z)$ can be evaluated at $z = 0$ by taking the limit as $z \to 0$ of their series expansions, thus obtaining

$$\mathcal{A}_0(z) = 1 + \frac{2i}{\pi}\gamma, \qquad \mathcal{A}_1(z) = 0, \qquad \mathcal{A}_2(z) = \frac{1}{2} + \frac{i}{\pi}(\gamma - 1). \tag{A.9}$$

## A.2 Smoothness of $sgn(t-\tau)\frac{dr(t,\tau)}{dt}$

Recalling the parametrization $x = x(t) = (t, f(t))$, $y = y(\tau) = (\tau, f(\tau))$, and the definition of the distance $r(t,\tau) = \sqrt{(t-\tau)^2 + (f(t)-f(\tau))^2}$,

$$\begin{aligned}\frac{dr(t,\tau)}{dt} =& \frac{(x_1(t)-y_1(\tau))x_1'(t) + (x_2(t)-y_2(\tau))x_2'(t)}{r(t,\tau)} \\ =& \frac{(t-\tau)+(f(t)-f(\tau))f'(t)}{r(t,\tau)}.\end{aligned} \tag{A.10}$$

This function is smooth except possibly at $\tau = t$, when $r(t,\tau) = 0$. However, the limit $\tau \to t$ of its square exists, as can be shown using l'Hopital's rule:

$$\begin{aligned}\lim_{\tau\to t}\left[\frac{dr(t,\tau)}{dt}\right]^2 =& \lim_{\tau\to t}\frac{[(t-\tau)+(f(t)-f(\tau))f'(t)]^2}{(t-\tau)^2+(f(t)-f(\tau))^2} \\ \overset{l'H}{=}& \lim_{\tau\to t}\frac{[(t-\tau)+(f(t)-f(\tau))f'(t)][-1-f'(\tau)f'(t)]}{-(t-\tau)-(f(t)-f(\tau))f'(\tau)} \\ \overset{l'H}{=}& \lim_{\tau\to t}\frac{[-1-f'(\tau)f'(t)]^2+[(t-\tau)+(f(t)-f(\tau))f'(t)][-f''(\tau)f'(t)]}{1+f'(\tau)^2-(f(t)-f(\tau))f''(\tau)} \\ =& \frac{[1+f'(t)^2]^2}{1+f'(t)^2} \\ =& 1+f'(t)^2.\end{aligned} \tag{A.11}$$

Thus

$$\lim_{\tau\to t}\frac{dr(t,\tau)}{dt} = \pm\sqrt{1+f'(t)^2} = \pm|x'(t)|. \tag{A.12}$$

Since $\frac{dr(t,\tau)}{dt} < 0$ when $t < \tau$ and $\frac{dr(t,\tau)}{dt} > 0$ when $t > \tau$, thus

$$\lim_{\tau\to t} sgn(t-\tau)\frac{dr(t,\tau)}{dt} = |x'(t)|. \tag{A.13}$$

## A.3 Smoothness of $M(t,\tau)$ term in $K_{8,2}(t,\tau)$ kernel

Recall

$$M(t,\tau) = \frac{-i}{d}\left[\frac{dr(t,\tau)}{dt}\frac{-i}{\delta r(t,\tau)} - \frac{e^{i\delta t}}{e^{i\delta t}-e^{i\delta\tau}}\right],$$

where $\delta = \frac{2\pi i}{d}$. Thus

$$\begin{aligned}
\lim_{\tau\to t} idM(t,\tau) =& \lim_{\tau\to t} \frac{(t-\tau)+(f(t)-f(\tau))f'(t)}{\delta r^2(t,\tau)} - \frac{e^{\delta t}}{e^{\delta t}-e^{\delta\tau}} \\
=& \lim_{\tau\to t} \frac{\left[(t-\tau)+(f(t)-f(\tau))f'(t)\right](e^{\delta t}-e^{\delta\tau}) - \delta e^{\delta t} r^2(t,\tau)}{\delta r^2(t,\tau)(e^{\delta t}-e^{\delta\tau})} \\
\overset{l'H}{=}& \lim_{\tau\to t} \frac{\left\{\begin{array}{r} -\left[1+f'(\tau)f'(t)\right](e^{\delta t}-e^{\delta\tau}) - \delta\left[(t-\tau)+(f(t)-f(\tau))f'(t)\right]e^{\delta\tau} \\ +2\delta e^{\delta t}\left[(t-\tau)+(f(t)-f(\tau))f'(\tau)\right] \end{array}\right\}}{-2\delta(e^{\delta t}-e^{\delta\tau})\left[(t-\tau)+(f(t)-f(\tau))f'(\tau)\right] - \delta^2 r^2(t,\tau)e^{\delta\tau}} \\
\overset{l'H}{=}& \lim_{\tau\to t} \frac{\left\{\begin{array}{r} -f''(\tau)f'(t)(e^{\delta t}-e^{\delta\tau}) - 2\delta e^{\delta\tau}\left[1+f'(\tau)f'(t)\right] \\ -\delta^2 e^{\delta\tau}\left[(t-\tau)+(f(t)-f(\tau))f'(t)\right] - 2\delta e^{\delta t}\left[1+f'(\tau)^2-(f(t)-f(\tau))f''(\tau)\right] \end{array}\right\}}{\left\{\begin{array}{r} 2\delta\left[1+f'(\tau)^2-(f(t)-f(\tau))f''(\tau)\right](e^{\delta t}-e^{\delta\tau}) \\ +4\delta^2\left[(t-\tau)+(f(t)-f(\tau))f'(\tau)\right]e^{\delta\tau} - \delta^3 r^2(t,\tau)e^{\delta\tau} \end{array}\right\}} \\
\overset{l'H}{=}& \lim_{\tau\to t} \frac{\left\{\begin{array}{r} -f'''(\tau)f'(t)(e^{\delta t}-e^{\delta\tau}) + 3\delta f''(\tau)f'(t)e^{\delta\tau} \\ +3\delta^2 e^{\delta\tau}\left[1+f'(\tau)f'(t)\right] - \delta^3 e^{\delta\tau}\left[(t-\tau)+(f(t)-f(\tau))f'(t)\right] \\ -2\delta e^{\delta t}\left[3f'(\tau)f''(\tau)-(f(t)-f(\tau))2f'''(\tau)\right] \end{array}\right\}}{\left\{\begin{array}{r} 2\delta(e^{\delta t}-e^{\delta\tau})\left[3f'(\tau)f''(\tau)-(f(t)-f(\tau))f'''(\tau)\right] \\ -6\delta^2 e^{\delta\tau}\left[1+f'(\tau)^2-(f(t)-f(\tau))f''(\tau)\right] \\ +6\delta^3 e^{\delta\tau}\left[(t-\tau)+(f(t)-f(\tau))f'(\tau)\right] - \delta^4 r^2(t,\tau)e^{\delta\tau} \end{array}\right\}} \\
=& \frac{e^{\delta t}\left[-3i\delta^2(1+f'(t)^2)+3i\delta f''(t)f'(t)\right]}{6i\delta^2 e^{\delta t}\left[1+f'(t)^2\right]} \\
=& \frac{-1}{2}\left[1-\frac{f''(t)f'(t)}{\delta(1+f'(t)^2)}\right] = \frac{-1}{2}\left[1-\frac{d}{2\pi i}\frac{f''(t)f'(t)}{1+f'(t)^2}\right].
\end{aligned}$$

## A.4 Evaluating $I(c)$

In this section we derive the exact value of the following integral, which is the hypersingular part of $I_8(t)$:

$$I(c) = \int_{|z|=1} \frac{z^c}{1-z} dz. \tag{A.14}$$

Recall that the integral over the unit circle is to be understood as a Cauchy principal value at the point $z = 1$. As such, it can be computed as the limit as $\epsilon \to 0$ of the

complex integrals over the smooth paths

$$\gamma_\epsilon : \left[t - \frac{d}{2}, t + \frac{d}{2}\right] \to \{|z| = 1, |z-1| > \epsilon\}, \qquad \gamma_\epsilon(\tau) = e^{\frac{2\pi i}{d}(\tau - t)}. \tag{A.15}$$

In order to use Cauchy's theorem, define the smooth path $\gamma_1$ such that $\gamma_\epsilon + \gamma_1$ is closed:

$$\gamma_1 : \left[\frac{\pi}{2} + \theta_\epsilon, \frac{3\pi}{2} - \theta_\epsilon\right] \to \{|z-1| = \epsilon, |z| \le 1\}, \qquad \gamma_1(\tau) = 1 - \epsilon e^{-i\tau}, \tag{A.16}$$

where $\theta_\epsilon$ is such that $\gamma_1(\pi/2 + \theta_\epsilon)$ and $\gamma_1(3\pi/2 - \theta_\epsilon)$ lie on the unit circle, so $\theta_\epsilon \to 0$ as $\epsilon \to 0$. Moreover, the function $\frac{z^c}{1-z}$ is not analytic on any open ball around $z = 0$ (unless $c$ is an integer). Indeed $z^c = e^{c\log(z)} = |z|^c e^{ic\arg(z)}$ is only analytic on a slit complex plane, such as $\mathbb{C}\backslash\mathbb{R}^{\le 0}$. Thus the function $\frac{z^c}{1-z}$ is analytic inside the closed curve $\gamma = \gamma_\epsilon + \gamma_1 + \gamma_+ + \gamma_0 + \gamma_-$, where $\gamma_\epsilon$ and $\gamma_1$ are as before, and the other smooth paths are defined below:

$$\gamma_+ : [1, \epsilon] \to \{\epsilon \le |z| \le 1, \arg(z) = \pi\}, \qquad \gamma_+(\tau) = \tau e^{i\pi} \tag{A.17}$$

$$\gamma_- : [\epsilon, 1] \to \{\epsilon \le |z| \le 1, \arg(z) = -\pi\}, \qquad \gamma_-(\tau) = \tau e^{-i\pi} \tag{A.18}$$

$$\gamma_0 : [0, 2\pi] \to \{|z| = \epsilon\}, \qquad \gamma_0(\tau) = \epsilon e^{i(\pi - \tau)}. \tag{A.19}$$

Figure A.1 shows the final contour path used for the integration. To summarize, for any $\epsilon > 0, \epsilon < 1/2$,

$$I(c) = \lim_{\epsilon \to 0} \int_{\gamma_\epsilon} \frac{z^c}{1-z} dz \tag{A.20}$$

$$= \int_\gamma \frac{z^c}{1-z} dz - \lim_{\epsilon \to 0} \int_{\gamma - \gamma_\epsilon} \frac{z^c}{1-z} dz \tag{A.21}$$

$$= 0 - \lim_{\epsilon \to 0} \int_{\gamma_1 + \gamma_+ + \gamma_0 + \gamma_-} \frac{z^c}{1-z} dz. \tag{A.22}$$

The integral over $\gamma_0$ goes to 0, since its absolute value does:

$$\left| \int_{\gamma_0} \frac{z^c}{1-z} dz \right| = \left| \int_0^{2\pi} \frac{-i\epsilon^{c+1} e^{i(c+1)(\pi - \tau)}}{1 - \epsilon e^{i(\pi - \tau)}} d\tau \right| \tag{A.23}$$

$$\le \epsilon^{c+1} \int_0^{2\pi} \frac{|e^{i(c+1)(\pi-\tau)}|}{|1 - \epsilon e^{i(\pi-\tau)}|} d\tau \tag{A.24}$$

$$\le \epsilon^{c+1} \int_0^{2\pi} \frac{1}{|1-\epsilon|} d\tau = \frac{2\pi\epsilon^{c+1}}{|1-\epsilon|} \to 0 \quad \text{as } \epsilon \to 0, \tag{A.25}$$

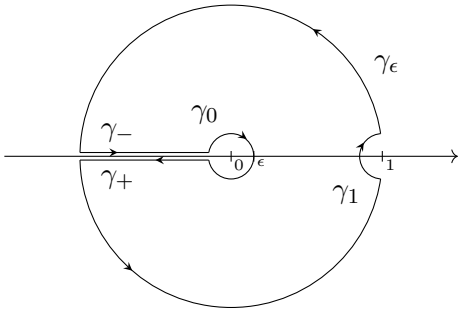



Figure A.1: Contour path used to compute complex integral $I(c)$.

since $c > -1$. The integral over $\gamma_1$ on the other hand can be computed by splitting the integrand in two:

$$\int_{\gamma_1} \frac{z^c}{1-z} dz = \int_{\gamma_1} \frac{1}{1-z} dz + \int_{\gamma_1} \frac{z^c - 1}{1-z} dz, \tag{A.26}$$

where $\frac{z^c-1}{1-z}$ is a smooth function, and thus can be bounded by a constant $M$ on $\gamma_1$. Since the length of the curve $|\gamma_1|$ goes to 0 as $\epsilon \to 0$, so does the integral of $\frac{z^c-1}{1-z}$ over $\gamma_1$, leaving only the integral of $\frac{1}{1-z}$ to compute:

$$\int_{\gamma_1} \frac{z^c}{1-z} dz = \int_{\gamma_1} \frac{1}{1-z} dz = \int_{\frac{\pi}{2}+\theta_\epsilon}^{\frac{3\pi}{2}-\theta_\epsilon} \frac{i\epsilon e^{-i\tau}}{\epsilon e^{-i\tau}} d\tau = i(\pi - 2\theta_\epsilon) \to i\pi \quad \text{as } \epsilon \to 0. \tag{A.27}$$

The integrals over $\gamma_+$ and $\gamma_-$ are very similar:

$$\int_{\gamma_+} \frac{z^c}{1-z} dz = \int_1^\epsilon \frac{\tau^c e^{ic\pi}}{1 - \tau e^{i\pi}} e^{i\pi} d\tau = -\int_\epsilon^1 \frac{\tau^c e^{i(c+1)\pi}}{1+\tau} d\tau \tag{A.28}$$

$$\int_{\gamma_-} \frac{z^c}{1-z} dz = \int_\epsilon^1 \frac{\tau^c e^{-ic\pi}}{1 - \tau e^{-i\pi}} e^{-i\pi} d\tau = \int_\epsilon^1 \frac{\tau^c e^{-i(c+1)\pi}}{1+\tau} d\tau, \tag{A.29}$$

and their sum becomes

$$\int_{\gamma_+ + \gamma_-} \frac{z^c}{1-z} dz = \left(e^{-i(c+1)\pi} - e^{i(c+1)\pi}\right) \int_\epsilon^1 \frac{\tau^c}{1+\tau} d\tau \tag{A.30}$$

$$= -2i \sin((c+1)\pi) \left( \int_0^1 \frac{\tau^c}{1+\tau} d\tau - \int_0^\epsilon \frac{\tau^c}{1+\tau} d\tau \right) \tag{A.31}$$

$$= -2i \sin((c+1)\pi) \left( \int_0^1 \frac{\tau^c}{1+\tau} d\tau - \epsilon^{c+1} \int_0^1 \frac{\tau^c}{1+\epsilon\tau} d\tau \right) \tag{A.32}$$

$$= -2i \sin((c+1)\pi) \left( \frac{\Gamma(c+1)\Gamma(1)}{\Gamma(c+2)} \, {}_2F_1(1, c+1; c+2; -1) \right.$$
$$\left. - \epsilon^{c+1} \frac{\Gamma(c+1)\Gamma(1)}{\Gamma(c+2)} \, {}_2F_1(1, c+1; c+2; -\epsilon) \right), \tag{A.33}$$

where ${}_2F_1(a_1, a_2; a_3; z)$ is the Gauss hypergeometric series defined in [67], and shown to be equal to its integral representation as long as $Re(a_3) > Re(a_2) > 0$, which is true here since $c > -1$. Note that ${}_2F_1$ is continuous at $z = 0$, so ${}_2F_1(1, c+1; c+2; -\epsilon) \to 0$ as $\epsilon \to 0$. Moreover, ${}_2F_1$ has a closed form in the particular case $z = -1$, leading to the following limit for the path integral as $\epsilon \to 0$:

$$\int_{\gamma_+ + \gamma_-} \frac{z^c}{1-z} dz \to -2i \sin\left[(c+1)\pi\right] \frac{\Gamma(c+1)\Gamma(1)}{\Gamma(c+2)} \frac{c+1}{2} \left[ \Psi\left(\frac{1}{2} + \frac{c+1}{2}\right) - \Psi\left(\frac{c+1}{2}\right) \right]$$
$$= -i \sin\left[(c+1)\pi\right] \left[ \Psi\left(\frac{c+2}{2}\right) - \Psi\left(\frac{c+1}{2}\right) \right],$$

where $\Psi$ is the digamma function.

Having computed the integrals over all the necessary smooth paths, the value of $I(c)$ is found to be

$$I(c) = -\lim_{\epsilon \to 0} \int_{\gamma_1 + \gamma_+ + \gamma_0 + \gamma_-} \frac{z^c}{1-z} dz \tag{A.34}$$

$$= -i\pi + i \sin((c+1)\pi) \left( \Psi\left(\frac{c+2}{2}\right) - \Psi\left(\frac{c+1}{2}\right) \right). \tag{A.35}$$

This can now be used to evaluate the integral $I_1^n(t)$, which was the objective of this appendix:

$$I_1^n(t) = \frac{de^{i\alpha_n t}}{2\pi i} \begin{cases} -i\pi + i \sin\left(\frac{\alpha_n d}{2}\right) \left[ \Psi\left(\frac{1}{2} + \frac{\alpha_n d}{4}\right) - \Psi\left(\frac{\alpha_n d}{4}\right) \right], & \text{if } n > \frac{-\alpha d}{2\pi} \\ i\pi - i \sin\left(\frac{\alpha_n d}{2}\right) \left[ \Psi\left(1 - \frac{\alpha_n d}{4}\right) - \Psi\left(\frac{1}{2} - \frac{\alpha_n d}{4}\right) \right], & \text{if } n \leq \frac{-\alpha d}{2\pi}. \end{cases} \tag{A.36}$$

**Numerical validation** To validate the expression in (A.35), a numerical scheme is used to approximate the integral over . For this purpose, $\gamma_\epsilon$ is parametrized as $\gamma_\epsilon(\theta) = e^{i\theta}$, where $\theta \in [-\pi, -\theta_\epsilon] \cup [\theta_\epsilon, \pi]$ and $\theta_\epsilon$ is such that $|1 - \gamma_\epsilon(\pm\theta_\epsilon)| = \epsilon$. Using this parametrization, the integral becomes

$$I(c) = \lim_{\epsilon \to 0} \int_{\gamma_\epsilon} \frac{z^c}{1-z} dz \tag{A.37}$$

$$= \lim_{\epsilon \to 0} \left\{ \int_{-\pi}^{-\theta_\epsilon} \frac{ie^{i(c+1)\theta}}{1-e^{i\theta}} d\theta + \int_{\theta_\epsilon}^{\pi} \frac{ie^{i(c+1)\theta}}{1-e^{i\theta}} d\theta \right\}. \tag{A.38}$$

The trapezoid method is used to approximate each of the two integrals, with $N$ number of points used for each, and evaluated for values of $\epsilon$ ranging from $10^{-1}$ to $10^{-30}$. Since there is a lot of cancellation between the two integrals, the symbolic package was used in Matlab, with the results being evaluated using 100 digits of accuracy.

**The case of integer $c$.** In this case, the exact formula (A.35) yields $I(n) = -i\pi$, and the numerical approximations also approach this value. Figure A.2 shows the norm of the numerical approximation as a function of $\epsilon$ for $c = 0$. As expected from the trapezoidal rule, the numerical convergence is second order linear. The values $c = 1, 2, 3$ were also tested, with identical results.

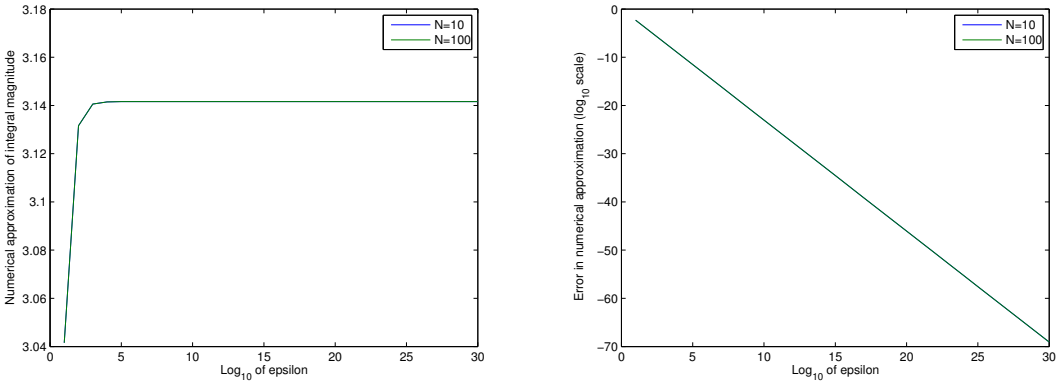



Figure A.2: Convergence of the numerical approximation to $I(0)$.

**The case $c = -1/2$.** In this case, the exact expression (A.35) yields $I(-1/2) = 0$. On the other hand, the numerical approximations are all very close to 0, as shown in figure

A.3, although the value is exponentially increasing with $\epsilon$.

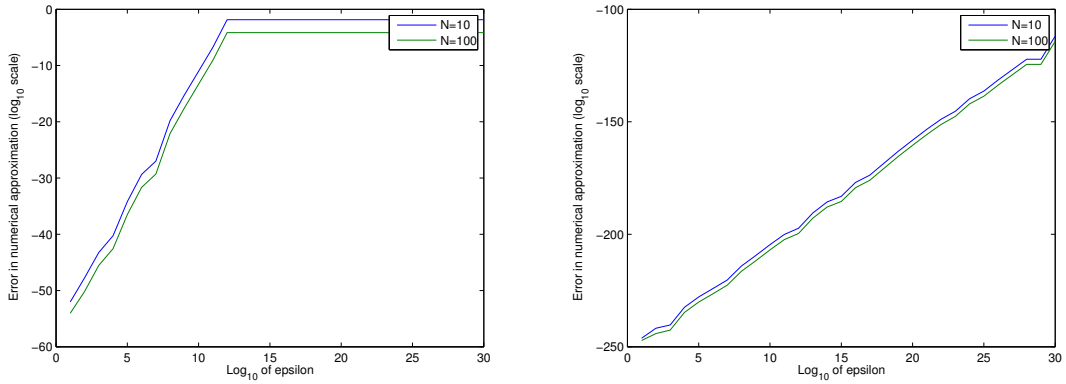



Figure A.3: Convergence of the numerical approximation to $I(-1/2)$, using respectively 16 and 100 digits of accuracy.

Note that the magnitude of the error, though increasing, is always well below the precision used, and decreases as the precision increases. This suggests that it is only due to round-off error from larger and larger cancellations as $\epsilon$ gets closer and closer to 0. Indeed, looking at the parametrization of $I(-1/2)$

$$I(-1/2) = \lim_{\epsilon \to 0} \left\{ \int_{-\pi}^{-\theta_\epsilon} \frac{ie^{i\theta/2}}{1 - e^{i\theta}} d\theta + \int_{\theta_\epsilon}^{\pi} \frac{ie^{i\theta/2}}{1 - e^{i\theta}} d\theta \right\}, \tag{A.39}$$

it is clear that the two integrals cancel each other exactly for any $\theta_\epsilon$, since the integrand is anti-symmetric:

$$f(\theta) = \frac{ie^{i\theta/2}}{1 - e^{i\theta}} = \frac{ie^{i(-1/2)(-\theta)}}{1 - e^{i(-1)(-\theta)}} = \frac{ie^{i(1/2)(-\theta)}}{e^{i(-\theta)} - 1} = -f(-\theta). \tag{A.40}$$

Thus as long as the numerical scheme is symmetric around 0, the numerical approximation will be exactly 0 for any $\theta_\epsilon$.

**The case of non-integer $c \neq -1/2$.** Choosing $c = 1/2$, the exact expression (A.35) yields $I(1/2) = -4i$. The numerical approximations approach the same value, and the convergence is linear up to a certain point, as shown in figure A.4. Similar results were obtained with $c = 5/2, 3/2, \pm 1/3, \pm 1/10, \pm 1/100$.

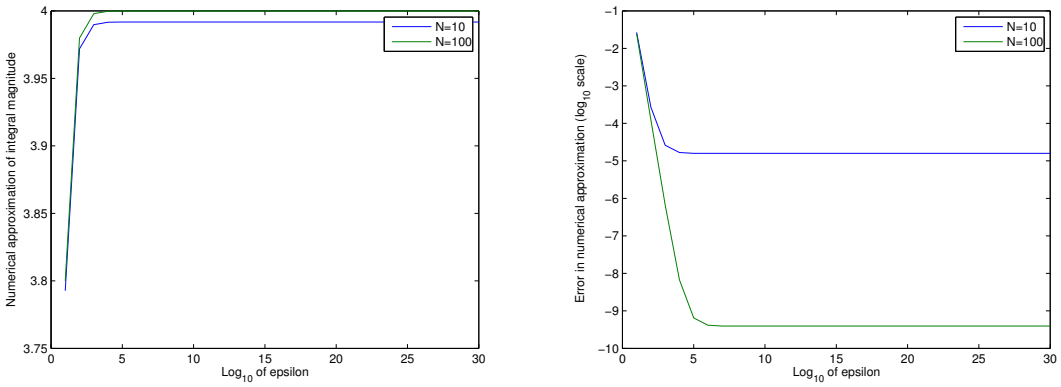



Figure A.4: Convergence of the numerical approximation to $I(1/2)$.

## A.5 Guide to Variable Names

| Text variable | Description |
|---|---|
| $a_n$ | Floquet coefficients of $u$ |
| $b_n$ | Floquet coefficients of $\frac{\partial u}{\partial \mathbf{n}}$ |
| $b_r$ | Rayleigh coefficients of $u^s$ |
| $d$ | Period of the surface profile |
| $e$ | ———————————- |
| $f(t)$ | Function describing the profile of the surface |
| $f_n^1$ | Floquet coefficients of $\xi_1$ |
| $f_n^2$ | Floquet coefficients of $\xi_2$ |
| $g_n^1$ | Floquet coefficients of $n_1\xi_1$ |
| $g_n^2$ | Floquet coefficients of $n_2\xi_1$ |
| $h_n$ | Floquet coefficients of $\xi_1'$ |
| $h_p$ | $p^{th}$ harmonic number |
| $i$ | ———————————- |
| $I_j$ | Integral corresponding to kernel $K_j$ |
| $k_i$, $k_e$ | Interior and exterior wave numbers |
| $K_j$ | Kernel used in integral $I_j$ |
| $K_{j,1}, K_{j,2}$ | Smooth parts of kernel $K_j = K_{j,1}\log\lvert t-\tau\rvert + K_{j,2}$ |
| $\mathcal{L}$ | Compact integral operator from integral system for solution $u$ |
| $\mathbf{n} = (n_1, n_2)$ | Normal to the surface |
| $r(t,\tau)$ | Euclidean distance between surface points $x(t)$ and $x(\tau)$ |
| $t$,$\tau$ | Parameters for the surface points $x$,$y$ |
| $u$, $u^s$, $u^{inc}$, $u^t$ | Total, scattered, incident and transmitted fields |
| $x$,$y$ | Points on the surface |