

NONLINEAR SURFACE PLASMON POLARITONS: ANALYTICAL AND
NUMERICAL STUDIES

by

YAN GUO

A DISSERTATION

Presented to the Department of Physics
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

March 2012

DISSERTATION APPROVAL PAGE

Student: Yan Guo

Title: Nonlinear Surface Plasmon Polaritons: Analytical and Numerical Studies

This dissertation has been accepted and approved in partial fulfillment of the requirements for the Doctor of Philosophy degree in the Department of Physics by:

Dr. Steven van Enk	Chair
Dr. Miriam Deutsch	Advisor
Dr. Jens Nöckel	Inside Member
Dr. Dietrich Belitz	Inside Member
Dr. Andrew Marcus	Outside Member

and

Kimberly Andrews Espy	Vice President for Research and Innovation / Dean of the Graduate School
-----------------------	--

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded March 2012

©March 2012

Yan Guo

DISSERTATION ABSTRACT

Yan Guo

Doctor of Philosophy

Department of Physics

March 2012

Title: Nonlinear Surface Plasmon Polaritons: Analytical and Numerical Studies

This dissertation contains analytical and numerical studies of nonlinear surface-plasmon polaritons (SPPs). In our studies, we consider SPP propagation at the interface between a noble metal with a cubic optical nonlinearity and an optically linear dielectric.

We first consider a sum-frequency generation process during the nonlinear interaction, where a nonlinear polarization with tripled frequency is generated from the incident fundamental SPP. Using the non-depletion approximation, the solution of the nonlinear wave equation shows a third harmonic generation process from the incident SPP wave. The solution is bound in the dielectric while freely propagating in the metal. For realistic noble metals with absorption, we use silver for its transparency window around the plasma frequency. In this window, absorption losses are reduced and the resultant signal has a good transmittance within the metal. The energy conversion efficiency from the incident SPP wave to the THG

signal is about 0.1% for excitation using a standard continuous wave laser with visible light intensity $I = 10^3 W/cm^2$. Once generated, the propagation angle of the signal is fully determined by the optical properties of the dielectric and the metal layers.

We next consider a nonlinear polarization with the same frequency as the incident light. In this process the third order nonlinearity of the metal is described by a nonlinear optical refractive-index. With the slowly varying amplitude approximation, the nonlinear wave equation takes the form of a nonlinear temporal Schrödinger (NLS) equation. The solution to the NLS equation for the nonlinear SPP is a temporal dark soliton (TDS). In addition to analytical studies, computational methods are also used. With no metal loss, the numerical solution shows stable propagation of a TDS, when the initial pulse has a *tanh* envelope satisfying the threshold peak amplitude. For an arbitrary input pulse, instabilities such as background-oscillations and multi-peak breakups occur. With metal loss, the input optical pulse decays while maintaining a single pulse shape when the initial amplitude satisfies the same *tanh* envelope condition as in the lossless case. For an arbitrary pulse, background-oscillations or pulse-breakups occur after a short time of propagation.

This dissertation includes previously published and unpublished co-authored material.

CURRICULUM VITAE

NAME OF AUTHOR: Yan Guo

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, Oregon
University of Science and Technology of China, Hefei, P.R.China

DEGREES AWARDED:

Doctor of Philosophy in Physics, 2012, University of Oregon
Master of Science in Physics, 2005, University of Oregon
Bachelor of Arts in Physics, 2003, University of Science and Technology
of China

AREAS OF SPECIAL INTEREST:

Nonlinear Optics, Nonlinear Plasmonics, Analytical and Numerical
Modeling, Numerical Computation and Simulations

PROFESSIONAL EXPERIENCE:

Graduate Research Assistant,
University of Oregon, 2005 – 2011
Graduate Teaching Fellow,
University of Oregon, 2003 – 2005

GRANTS, AWARDS AND HONORS:

Travel Grant, Division of Laser Science, American Physical Society (APS) 2010

Ph.D. Qualifying Exam First Place Award, Department of Physics, University of Oregon 2004

PUBLICATIONS:

Yan Guo, Miriam Deutsch, “Third Harmonic Generation using Surface Plasmon Polaritons at Nonlinear Interfaces”, *Frontiers in Optics (FiO)/Laser Science XXVI (LS) conference*, 696, FWN5 (2010)

Yan Guo, Miriam Deutsch, “Nonlinear Surface Plasmon Polariton Propagation and Third Harmonic Generation at a Planar Metal/Dielectric Interface”, *APS March Meeting Z15*, (2010).

ACKNOWLEDGEMENTS

My deepest gratitude is to my advisor, Dr. Miriam Deutsch, who was a co-author of all the materials presented in this dissertation. I have been feeling amazingly fortunate to have an advisor who gave me guidance and encouragement during the study and research of all these years. Her patience and kindness, as well as her enthusiasm in science and sound advice in research, have been invaluable to me. This dissertation would not have been possible without her efforts and help.

I am deeply grateful to Dr. Peter Hugger who has provided assistance in numerous ways. Peter has been my English guru, good roommate, and friend for many years. Its a great pleasure to know him in and out of physics. I am also thankful to him for his endured efforts in carefully reading and editing on countless revisions of this dissertation.

I am indebted to many of my colleagues to support me, especially Lawrence Mick Davis. I really cherish the countless discussions we had in research and exchange of ideas. Its a great honor to work with him.

I wish to thank all my classmates in grad-school, Andrew, Cody, Jeremy, Matt, Pete, Libby, Emelie and Kathy, for the study time with happiness and hardness, for the entertainment, and for the friendship. I wish to thank my Chinese friends in the Physics Department, Yupeng, Xianghui, Yan, Jun and Xiaolu, for making Eugene

home. I also want to thank my friends in the Physics Department, Shannon, Billy, Chuck, Mary and Carey, for helping me get through the difficult times.

This work could not have been completed without the support of the committee: Dr. Steven van Enk, Dr. Jens Nöckel and Dr. Dietrich Belitz in Department of Physics, and Dr. Andy Marcus in Department of Chemistry. I also would like to thank Dr. Susanta Sarkar for teaching me experimental skills; Dr. Keisuke Hasegawa for the initial momentum on computer simulations and modelings; and Bonnie Grimm and other staff members of the physics office and the Oregon Center for Optics for backing me up inside the Willamette Hall.

Finally I would like to thank my wife, Chu Chen. She is always giving me the strongest support and greatest understanding. Thank you for always being there for me.

To CHEN Chu and GUO Rui.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
From Electronics and Photonics to Plasmonics	1
Organization of This Dissertation	10
II. INTRODUCTION TO PLANAR SURFACE PLASMON POLARITONS	12
Optical Properties of Noble Metals	12
Planar Surface Plasmon Polaritons	18
III. INTRODUCTION TO NONLINEAR MEDIA AND NONLINEAR WAVE EQUATIONS	30
Introduction to Nonlinear Optics	30
Wave Equations in a Nonlinear Medium	32
Simplifications of the Nonlinearities in an SPP System	38
IV. HARMONIC GENERATION IN SURFACE PLASMON POLARITONS	43
Theoretical Development of Third Harmonic Generation of Nonlinear SPPs	43
Conversion Efficiency and Propagation Angles	51
Adjustment to Realistic Noble Metals	55
Conclusions	57
V. PLASMONIC SOLITONS	59

Chapter	Page
Introduction to Optical Solitons.....	59
The Temporal Nonlinear Schrödinger Equation of SPPs	66
Nonlinear SPP Dispersion Relations.....	73
Analytical and Numerical Solutions of Dark Solitons.....	77
Numerical Analysis with and without Metal Loss	82
Conclusions	88
VI. CONCLUSIONS.....	89
APPENDICES.....	95
A. SOLUTION OF INHOMOGENEOUS SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS	95
B. HELMHOLTZ THEOREM AND HELMHOLTZ EQUATION	97
C. THE ROTATION MATRIX	99
D. NUMERICAL ALGORITHM FOR NLS EQUATIONS	113
E. C/C++ SOURCE CODE FOR FDTD METHOD	124
FDTDmain.cpp.....	124
FDTDarray.h.....	167
FDTDarray.cpp	169
FDTDcubic.h	176
FDTDcubic.cpp	180
FDTDtime.h	190
FDTDtime.cpp	191
REFERENCES CITED	194

LIST OF FIGURES

Figure	Page
1.1 Moore's sketch.	2
1.2 Plasmonics is the bridge between electronics and photonics.	4
1.3 A schematic of SPPs.	6
1.4 All-optical switching.	8
1.5 A schematic of a surface plasmon resonance biosensor.	9
2.1 The modified Drude model.	18
2.2 Sketch of an SPP-supporting interface.	19
2.3 The dispersion relation of SPPs	20
2.4 Prism coupling methods.	21
2.5 TE and TM mode separation.	25
2.6 The penetration depths and a snapshot of the SPPs.	29
3.1 Sum frequency generation.	34
3.2 Difference frequency generation.	35
3.3 Three coordinate systems.	40
4.1 A graph of the THG solution in vacuum.	49
4.2 A graph of the THG solution in metal.	49
4.3 A plot of the propagation angle.	53
4.4 Drude model vs realistic metals.	56
4.5 A plot of propagation angle with silver data.	57
5.1 Temporal Soliton.	61
5.2 Spatial Soliton.	62
5.3 A bright soliton and a dark soliton.	65
5.4 The coefficients in the NLS equation.	78
5.5 The stability of the soliton presented by numerical calculations.	83
5.6 Pulse break-up and oscillation.	84
5.7 The evolution of a dark SPP pulse with loss.	86

Figure	Page
5.8 Pulse break-up and oscillation.	87
C.1 The rotation matrix for <i>crystal</i> - <i>lab</i> coordinates conversion.	99
C.2 The relation between (111) crystal “cut” and the lab coordinates.	102
C.3 The relation between (110) crystal “cut” and the lab coordinates.	103
C.4 The relation between (100) crystal “cut” and the lab coordinates.	104
D.1 Crank-Nicolson scheme illustrated.	115

LIST OF TABLES

Table	Page
2.1 Maxwell's equations and units.....	13
2.2 Parameters in the modified Drude model.....	17
D.1 Numerical solution with lossless metals.	118
D.2 Numerical solution with lossy metals.....	119

CHAPTER I

INTRODUCTION

From Electronics and Photonics to Plasmonics

In the computer hardware industry, the well-known Moore's Law predicts the growth speed of chip technology. It states that the number of transistors that can be printed per square inch on a chip doubles approximately every 18 months, shown in Fig. 1.1.

This famous observation was made by Intel co-founder Gordon Moore in 1965 and has described the pace of silicon technology growth ever since then. Today, decades later, Moore's Law remains true. However Moore's Law is also meeting new challenges and slowing down [1]. The major challenges opposing further improvements are the electronic thermal effect (Joule heating) and signal speed delay within electronic interconnections [2]. The Joule heating effect is inherent for conductors carrying electric current, and is caused by the interactions between the moving electrons and the background atomic ions. The heating problem shows more dramatically when the integrated circuits enter the sub-micron dimension in size, for both more heat is generated per unit area and the heat is not able to dissipate effectively. The temperature built up by Joule heat in turn deteriorates the

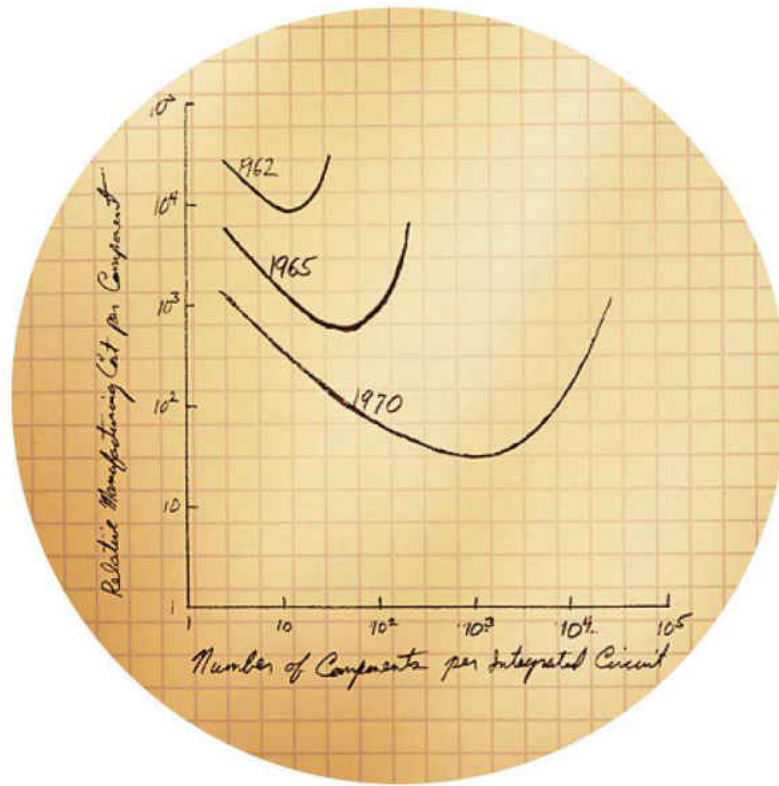


Figure 1.1 Moore's sketch. In 1965, Gordon Moore sketched out his prediction of the pace of silicon technology. It is still true today while facing new challenges. The picture is taken from http://download.intel.com/museum/Moores_Law/Printed_Materials/Moores_Law_2pg.pdf. The axis labels read: "Number of Components Per Integrated Circuit" (x-axis), and "Relative Manufacturing Cost/Component" (y-axis)

quality and the speed of the electronic interconnections. In a world where smaller is better, today's state-of-the-art micro chips shrink the dimension of circuit units to the order of $50nm$ [3] and the bottleneck to the development of smaller and

faster integrated circuit chips has become the fundamental conductivity properties of electrons [4]. For electronic integrated circuits, the limiting factors also include the distributed capacitance and the cross-talk between components factors which are fundamentally caused by the finite mass of electrons, preventing the production of electronic devices on yet smaller scales.

One of the possible solutions to the major problems in electronics is to use photonics technology. Photons provide great data carrying capacity, and high speed data processing photons operate at optical frequencies and travel at the speed of light. Since direct interactions between photons are extremely weak, and negligible in practice, photonics devices are robust to environmental photonic noise. In addition, for a given data transmission rate, photonics devices exhibit much lower power consumption than electronic devices and lose less energy as heat. Because of these strengths in photonics, the invention of laser and fiber communication technology has greatly enhanced information processing speed and accurate long distance information relay. Presently, photonics is being investigated as a method to connect electronic devices within sub-micron distance on a chip, as illustrated in Fig. 1.2.

One of the fundamental limiting factors in photonics, however, is the diffraction limit of optical waves. The typical operating wavelength in telecom optical fibers is about $1.5\mu m$ [5], which sets the lower limit of optical devices and operation dimension to be no smaller than hundreds of nanometers. More precisely, when the

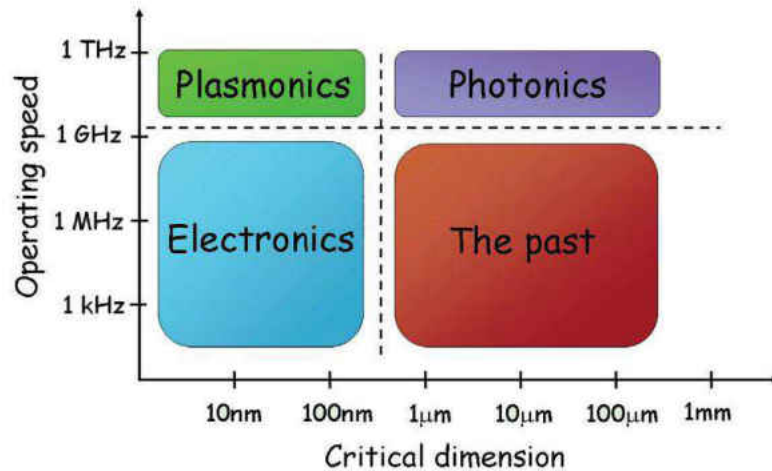


Figure 1.2 Plasmonics is the bridge between electronics and photonics. It utilizes metal/dielectric interfaces as optical wave guides and confines optical waves to subwavelength dimensions. The operating speed of plasmonic devices is still determined by the speed and bandwidth of light waves. Thus it takes advantage of operating speeds of photonic devices and the small critical dimensions of electronics devices. (Figure taken from ref. [4]).

operating wavelength of the light is $\lambda \sim 1500nm$, the wave nature of the photons gives the operation dimension, d , its lower bound according to Abbe diffraction limit of light waves, $d = \lambda/2n$ where n is the optical refractive index of the medium in which the light travels. For example in glass, the operating dimension is roughly $530nm$ by the Rayleigh criterion since glass has a refractive index of $n = 1.5$. For photons, the ability to store and transport data deteriorates when the device size scale is under its half wavelength. Therefore photonic devices are at least one order of magnitude larger than nano-scale electronic components.

In order to facilitate both optical and electronic devices integrated onto the same chip, a bridge of interfacing needs to be built to close the gap between the two very different technologies. This bridge is afforded by the photonics-electronics hybrid, i.e. plasmonics (see Fig. 1.2).

The fundamental element of plasmonics is the surface plasmon-polariton (SPP). In 1956 Pines described oscillations of the free electron gas in a conducting metal as “plasmons”, in analogy to earlier study on plasma oscillations in gas discharges [6]. The term “polariton” was first used by Fano in the same year to describe the particle-like coupled oscillation formed by interactions between light and electrons in a metal [7]. Finally the term “surface plasmon-polariton” was coined by Cunningham and his colleagues in 1974 [8].

The interactions between light waves and the free electrons in metal lead to the SPP modes guided by the dielectric-metal interface. As a result, the electromagnetic wave is bound on the metal surface, with only evanescent fields into the dielectric and metallic media. Consequently, the planar metal surface acts as a waveguide for the surface wave and confines the electromagnetic fields to within about half a wavelength of the surface. Even though the penetration depth in the metal is about $10nm$ for SPPs, the propagation distance of the SPP is limited by a gradual energy dissipation from the SPP into the metal, as shown in Fig. 1.3. A typical SPP propagation length in silver or gold is on the order of microns, and can be

extended to millimeters with proper structural preparation of the supporting metal [3] [9].

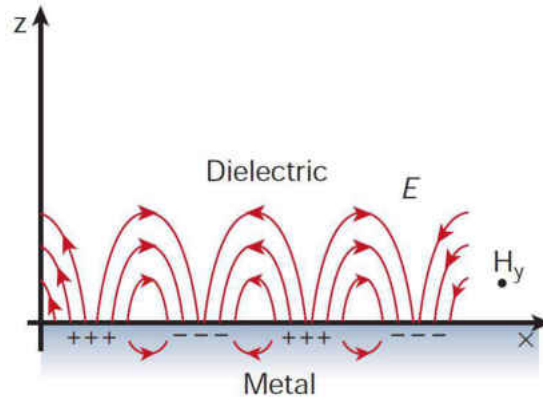


Figure 1.3 A schematic of SPPs. SPPs at the interface between a metal and a dielectric material formed from electromagnetic wave and surface charge interactions. The magnetic field is transverse only (H is in the y direction) while the electric field has both in-plane (x and z) components. The wave can propagate in the x direction for a distance of more than ten wavelengths before the energy is dissipated in the metal. (Figure taken from ref. [10]).

The simplest plasmonics are constructed by an electromagnetic wave propagating along a vacuum-metal interface as illustrated in Fig. 1.3. For more complicated applications with plasmonics, however, more complicated interface geometries are built for different purposes. For example, optical wave guides using metal nanostructures have sub-wavelength critical dimension, and thus allow further reduction of the device size to below the diffraction limit of photonics in a variety of geometries [11] [12].

The scientific branch of plasmonics has been developing rapidly for decades. Numerous applications and many variations of plasmonics design have been explored and researched for more than fifty years. For example, a dielectric-metal-dielectric structured stripe waveguide was designed for thermo-optic Mach-Zender interferometric modulators and directionally-coupled switches, which utilize long-range SPPs guided by the metal stripe [3] [13] [14] [15] [16]. In order to guide SPPs around curvatures such as corners [12] [17] [18] without losing high efficiency in energy transmission and switching, arrays of closely spaced metal nanoparticles were proposed, where the arrays were used for coupled plasmon modes between the metal particles [19]. Similarly, ultracompact modulators have been revolutionized using plasmonics. Modulators with femtojoule switching energies and gigahertz modulation frequencies were demonstrated using a field-effect modulation of plasmon waveguide modes in a metal-oxide-semiconductor geometry [20] [21]. Additionally, ultrafast, broad bandwidth optical switching of SPPs has been demonstrated. The all-optical switching of a propagating SPP signal can be built on a metal-dielectric waveguide with direct optical modulations by optical excitation of the metal as shown in Fig. 1.4 [22] [23] [24] .

The evanescent waves in the transverse direction of the SPPs, on the other hand, provide sub-micron detecting sensitivities, and thus become a very important method in gas detection, molecular optical chemical sensing and biosensing [25] [26] [27] [28] [29]. These sensors operate based on surface plasmon resonance (SPR)

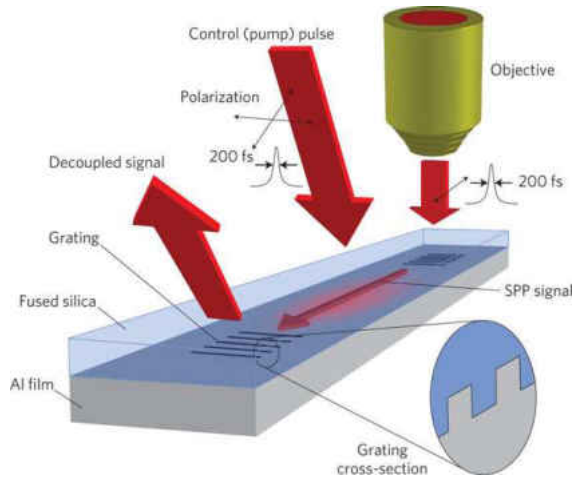


Figure 1.4 A schematic showing of ultrafast all-optical switching of SPP waves along an aluminium/silica interface. The switching is modulated by optical pump pulses. The set up uses the grating-coupling method to excite SPPs from a laser and to decouple signals to detectors. (Figure taken from ref. [22]).

effects. They have the great advantage of highly sensitive to the variation of the molecule layer and fast enough for real-time detection. These sensors also have the great advantage of being "label-free", as they can function without a label on a molecule (such as radioactivity or fluorescence). One layout of such sensors is shown in Fig. 1.5. It positions the evanescent waves of the SPP excitation on top of a metal film on the light-incident side and the optical detector underneath. The binding of molecules, such as O_2 , CO_2 or DNA to the film surface changes the optical refractive index in the immediate vicinity of the surface layer, and produces an easily observable shift of the resonance angle of the incident light source [25] [30] [31].

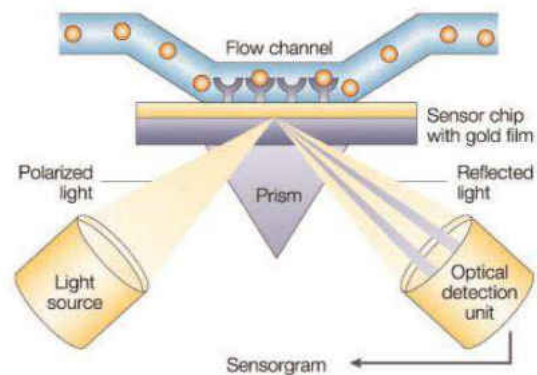


Figure 1.5 A schematic of a surface plasmon resonance biosensor. This biosensor functions by monitoring the angle shift of minimum light reflection. The biosensor operates at the critical angle that the surface plasmon resonance (SPR) condition is satisfied and the reflection angle of the light from the surface is monitored. The reflection angle shifts due to the change of the optical index on the flow channel side by the amount of molecules attached at the surface. (Figure taken from ref. [30]).

The strong confinement of the SPPs' electromagnetic fields can greatly enhance the nonlinearity of the host medium, and so the study of SPPs quickly becomes also the study of nonlinear optics. Such nonlinear optical effects include surface enhanced Raman scattering (SERS) when the localized SPP frequency is in resonance with the incident laser frequency [32] [33]. In addition, the surface enhanced harmonic generations, especially the second harmonic generations which arise from the asymmetric surface profile of the SPP structure, are also observed and demonstrated [34] [35] [36].

Organization of This Dissertation

This dissertation is organized as follows. In Chapter II we first review the optical properties of the noble metals. Based on the optical properties of the noble metals, the surface plasmon polaritons (SPPs) are introduced and derived from Maxwell's equations. The attributes of the linear SPPs are used and expanded in later chapters. The material in this chapter is a review of several textbooks and previous published articles by other authors, such as [37] and [38].

Chapter III gives a general introduction to nonlinear optics, especially the case with third order susceptibility $\chi^{(3)}$. The third order nonlinearity has two major branches: sum frequency generation and nonlinear optical index. Those two branches lead to our research interest as in SPP harmonic generation (Chapter IV) and plasmonic solitons (Chapter V). This chapter is a review of several textbooks such as [39] [40] and [41], with original work in theoretical derivation of nonlinear wave equations and nonlinear polarization that are specified to our case of study, which is previously coauthored and published with Miriam Deutsch in [42].

In Chapter IV, a solution of the nonlinear wave equations is given. The steady-state solution shows a third harmonic generation (THG) induced by the incident SPP and can radiate through the metal. For realistic metals, experimental data adapted from the literature are used. The realistic metal is chosen to be silver because silver has a transparency window around its plasma frequency and gives good transmittance for the induced THG wave. The energy conversion efficiency

from the incident SPP to the THG is calculated and shown to be consistent with the assumptions and approximations made for the original nonlinear wave equations. The propagation angle for the THG is also derived, using both an analytical model of the metals and experimental data from the literature. The material in this chapter was previously coauthored and published with Miriam Deutsch in [42] and [43].

In Chapter V, we follow established approaches for our nonlinear SPP system and achieve a temporal nonlinear schrödinger (NLS) equation. The solution to this NLS equation gives a temporal dark soliton. Attributes such as stability is studied under scenarios of both lossless and lossy metal, with standard numerical methods and programming. The material in this chapter was previously coauthored with Miriam Deutsch and has not yet unpublished

CHAPTER II

INTRODUCTION TO PLANAR SURFACE PLASMON POLARITONS

Optical Properties of Noble Metals

The physics of plasmonics derives from electromagnetic modes called surface plasmon polaritons (SPPs), which arise from the interaction between light and mobile surface charges in the metal. The SPPs usually propagate along a conductor-dielectric interface and the conductors are commonly metals with high conductivities and low absorption such as Cu, Al and some noble metals (Ag and Au) [10] [44] [45].

Before we dive into the world of surface plasmon polaritons, the optical model of the main supporting medium, the noble metals, needs to be discussed in detail. In the field of classic electromagnetism, Maxwell's equations are the cornerstone of all phenomena and analysis [38]. We start our analysis with the macroscopic Maxwell's equations in the following form:

$$\nabla \cdot \mathbf{D} = \rho_{free} \quad (\text{II.1a})$$

$$\nabla \cdot \mathbf{B} = 0 \quad (\text{II.1b})$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (\text{II.1c})$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J}_{total} \quad (\text{II.1d})$$

We use SI units, see Tab.2.1.

Table 2.1 Maxwell's equations and units

Symbol	Name	Unit
E	Electric Field	V/m
B	Magnetic Field	T
D	Electric Displacement Field	C/m^2
H	Auxiliary Magnetic Field	A/m
ϵ_0	Vacuum Permittivity	F/m
μ_0	Vacuum Permeability	N/A^2

For the current density \mathbf{J}_{total} , the most general expression [39] in terms of multipole expansion is

$$\mathbf{J}_{total} = \mathbf{J}_c + \frac{\partial \mathbf{P}}{\partial t} + \nabla \times \mathbf{M} - \frac{\partial}{\partial t}(\nabla \cdot \mathbf{Q}) + \dots \quad (\text{II.2})$$

where \mathbf{P} is the electric dipole, \mathbf{M} is the magnetic dipole, \mathbf{Q} is the electric quadrupole, and so on. In the absence of sources, $\rho_{free} = 0$ the first term in (II.2), the conducting current density, disappears, $\mathbf{J}_c = 0$. In non-magnetic media, which react only slightly to magnetic fields, the magnetization \mathbf{M} in the third term of (II.2) is considered negligible in optical interactions. Consequently, the relative permeability of non-magnetic media, for our case the noble metals, is very close to one in $\mathbf{B} = \mu_0 \mu \mathbf{H}$ and we can safely use $\mu = 1$ in the context of optical properties of the noble metals. This leads to a simple linear relation between the magnetic induction and magnetic field: $\mathbf{B} = \mu_0 \mathbf{H}$. For terms related to the electric charge distribution in (II.2), the dipole term $\frac{\partial \mathbf{P}}{\partial t}$ dominates and the rest of the higher

order terms - quadrupole \mathbf{Q} and higher - are dropped. Thus the optical properties in a medium are determined to the first order of the electric field \mathbf{E} by the dipole polarization \mathbf{P} and are also represented in the dielectric displacement \mathbf{D} as

$$\mathbf{D} = \varepsilon_0 \mathbf{E} + \mathbf{P} \quad (\text{II.3})$$

The relationship between the dipole polarization \mathbf{P} and the electric field \mathbf{E} is macroscopically determined by the dielectric susceptibility, χ , via

$$\mathbf{P} = \varepsilon_0 \chi \mathbf{E} \quad (\text{II.4})$$

where χ is a tensor but is often simplified to a scalar quantity for isotropic media, such as noble metals.

In a certain frequency limit, usually around the ultraviolet frequency range, the optical response of the metal permittivity can be described by the plasma model, where a free electron gas with number density \bar{n} and charge e moves against positive background made of ion cores. In the plasma model, the electron mass m_e is corrected to an effective optical mass m due to the band structure of the medium [46]. The fact that most metals possess very high electric conductivities is a direct result of the plasma model and we can use the plasma model for most of our studies. In the plasma model, the conduction electrons in the metal can respond and follow the frequency of the external electric field up to a limit, and this cut-off frequency $\omega_p = \sqrt{\frac{\bar{n}e^2}{m\varepsilon_0}}$ is known as the plasma frequency. For alkali metals where the single conduction electron per primitive cell is simply treated as completely free, the

plasma frequency ω_p occurs above ultraviolet optical frequencies. For noble metals, on the other hand, due to the presence of d -bands, the plasma frequency is in the visible frequency range, where absorption from interband transitions dominates [46]. We take this effect into account in later chapters. For the purposes of this research, silver and gold are the best candidates to support SPPs. Silver and gold are members of the noble metals family. They are chosen for both their chemical stability and electric conductivity. Henceforth the “noble metals” referred to in this document are silver or gold unless otherwise stated.

The electric permittivity, or dielectric function, is the primary determining factor of optical properties of the noble metals. Here we follow the classic electron gas model for the dielectric function [47], known as the Drude-Lorentz model for dielectrics. The electrons in silver and gold take the effective mass, $m = 1.1m_e$ where m_e is the mass of an electron. They obey Newton’s second law of motion in the presence of an external electric field $\mathbf{E}(\mathbf{x}, t)$:

$$-e\mathbf{E} = m[\ddot{\mathbf{x}} + \gamma_p\dot{\mathbf{x}} + \omega_0^2\mathbf{x}] \quad (\text{II.5})$$

In the Drude model for metals, the restoring force is considered zero for mobile electrons such that $\omega_0 = 0$. The motion of the free electrons in the noble metals is damped via collisions with crystal impurities and is characterized by a phenomenological damping constant $\gamma_p = 1/\tau_{relax} \sim 100$ THz where τ_{relax} is the relaxation time of the free electron gas on the order of femtosecond at room temperature [46].

A harmonic solution $\mathbf{x}(t) = \mathbf{x}_0 e^{-i\omega t}$ can be written down for (II.5) with an electro-magnetic field of the form $\mathbf{E}(\mathbf{x}, t) = \mathbf{E}(\mathbf{x}) e^{-i\omega t}$, where \mathbf{x}_0 is a complex amplitude representing the phase shift between \mathbf{x} and \mathbf{E} . With the solution

$$\mathbf{x}(t) = \frac{e}{m(\omega^2 + i\gamma_p\omega)} \mathbf{E}(t) \quad (\text{II.6})$$

it is straightforward to show the macroscopic polarization has the form

$$\mathbf{P} = -Ne\mathbf{x} = -\frac{Ne^2}{m(\omega^2 + i\gamma_p\omega)} \mathbf{E} \quad (\text{II.7})$$

where \bar{n} is the electron density in metal. Using the expression for electric displacement (II.3) yields

$$\mathbf{D} = \varepsilon_0 \left(1 - \frac{\omega_p^2}{\omega^2 + i\gamma_p\omega} \right) \mathbf{E} \quad (\text{II.8})$$

where $\omega_p^2 \equiv \frac{Ne^2}{\varepsilon_0 m}$. So the dielectric function for a noble metal is

$$\varepsilon(\omega) = 1 - \frac{\omega_p^2}{\omega^2 + i\gamma_p\omega} \quad (\text{II.9})$$

The positive background of the ion cores in the metal induces a residual polarization for $\omega \rightarrow \infty$, therefore a dielectric constant ε_∞ is introduced to the Drude-Lorentz model and the dielectric function for metal is written as

$$\varepsilon_m(\omega) = \varepsilon_\infty - \frac{\omega_p^2}{\omega^2 + i\gamma_p\omega}. \quad (\text{II.10})$$

Empirically, gold and silver exhibit much stronger optical absorption than the Drude model predicts due to the effect of the actual band structures in the metals [48] [49] and the skin depth change in the measured experiment data [50] [51].

Consequently an effective absorption term $i\frac{\sigma}{\epsilon_0\omega}$ is added to better fit the model where σ is the effective AC conductivity at optical frequencies [52] [53]. In this dissertation, when absorption is important to the results, the modified Drude model is used

$$\epsilon(\omega) = \epsilon_\infty - \frac{\omega_p^2}{\omega^2 + i\gamma_p\omega} + i\frac{\sigma}{\epsilon_0\omega}. \quad (\text{II.11})$$

For silver, we mostly follow [54] and the parameters are listed in Table[2.2].

Table 2.2 Parameters in Modified Drude Model [37].

Symbol	Value	Unit
ϵ_∞	5.1	—
ω_p	8.6	eV
γ_p	45	meV
σ	23.35	$(\Omega \cdot \mu m)^{-1}$

The Drude model represents the optical properties of silver very well below the plasma frequency ω_p , whose wavelength equivalent is about 320nm for wavelength equivalence, but fails for frequencies above ω_p . A comparison of the real and imaginary parts of ϵ taken from both data and the theoretically predicted values given by the modified Drude model is given in Fig. 2.1. The experimental data and theoretical results from the Drude model agree well over all wavelengths except those approaching the plasma frequency. The difference between the experimental data and the Drude model will be discussed in Ch.IV.

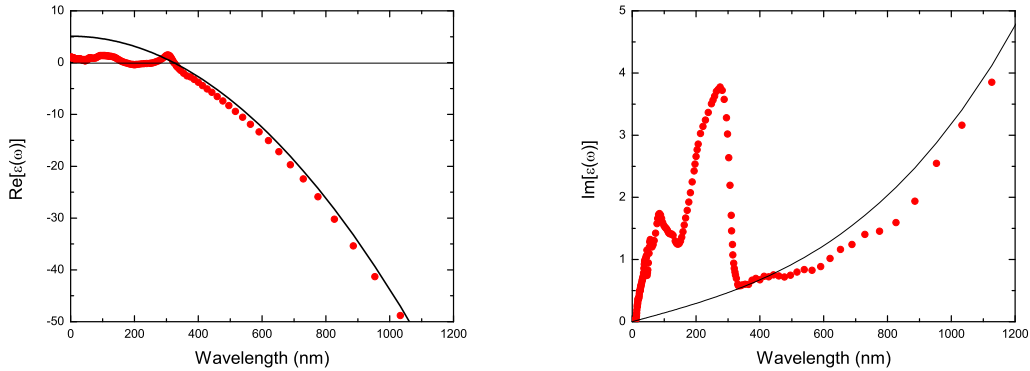


Figure 2.1 The modified Drude model (lines, from (II.11)) vs experimental data (dots, taken from [37]) for silver. For wavelengths greater than 320nm , the Drude model correctly represents the optical properties of silver. For wavelengths shorter than 320nm , however, the increasing absorption due to inter-band transition dominates and the Drude model is invalid.

Planar Surface Plasmon Polaritons

Introduction

The supportive structure of surface plasmon polaritons requires at least one interface formed by a metal and a dielectric, as shown in Fig. 2.2. The y direction is extended to infinity in our model so that the planar SPPs discussed here are essentially two-dimensional, with only x and z dependence.

The SPPs, which are EM waves that are guided by a metal surface, have a larger propagation wavevector than the wavevector in vacuum for the same frequency. Therefore additional momentum is required to excite an SPP wave

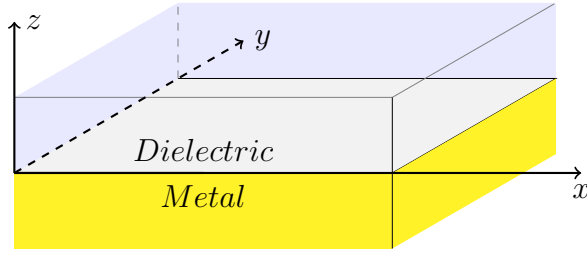


Figure 2.2 Sketch of an SPP-supporting interface. \mathbf{x} is the direction of propagation, \mathbf{z} is the normal to the interface pointing from the metal to the dielectric, and \mathbf{y} is the transverse direction extended to infinity.

along a metal surface as seen in Fig. 2.3. There are two classical experimental configurations used to excite the SPP from light in free space, the Kretschmann configuration and the Otto configuration (see Fig. 2.4). Both configurations utilize the fact that the wavevector of light is larger in a high index prism than in free space, so when the total internal reflection condition is satisfied, the evanescent wave tunnels out and excites SPPs on the metal surface.

There are also ways to excite SPPs without a high index prism, such as by scattering light over a rough surface, or by using manufactured gratings to match the missing momentum between light in free space and the SPP on a planar metal surface, as shown in Fig. 1.4. Recently it was also demonstrated that one may excite the SPP directly from free space by optical nonlinearities in the SPP supporting dielectric or metal [55] [56].

The simplest geometry layout that supports surface plasmons is shown in Fig. 2.2. The surface plasmon polaritons supported by this layout are therefore

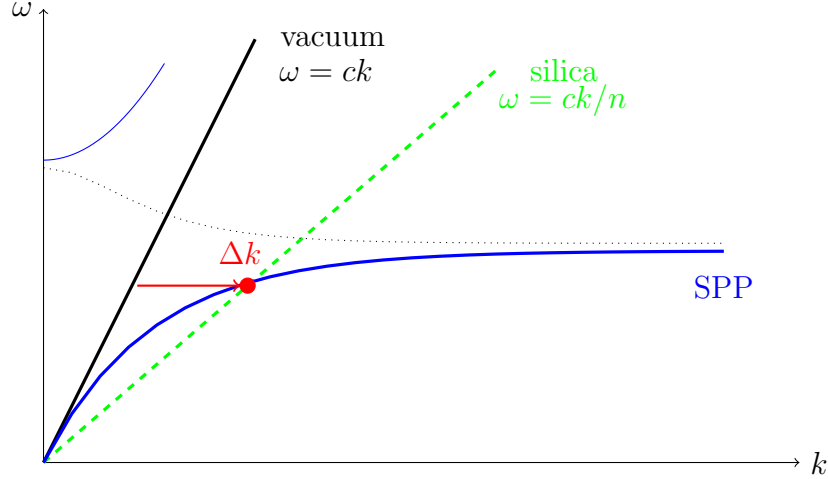


Figure 2.3 The dispersion relation of SPPs. There is a momentum mismatch (red arrow) between the SPP wavevector (blue solid curve) and the vacuum light line (black solid line). Therefore, some special techniques are needed to compensate for the missing momentum, such as using a high index prism (green dashed line) to alter the light line to intersect (red dot) with the SPP dispersion curve.

called planar SPPs, differentiating from SPPs supported by other geometrical structures such as nanoshells [57]. In photonics and plasmonics, the planar surface layout is frequently used in chemical and biological applications such as biosensing and nanoscale waveguides [27] [29] [30] [25].

Derivation of SPPs Using Maxwell's Equations

We start the discussion with the Maxwell equations:

Combining the two curl equations (II.1c) and (II.1d) gives

$$\nabla \times \nabla \times \mathbf{E} = -\mu_0 \frac{\partial^2 \mathbf{D}}{\partial t^2} \quad (\text{II.12})$$

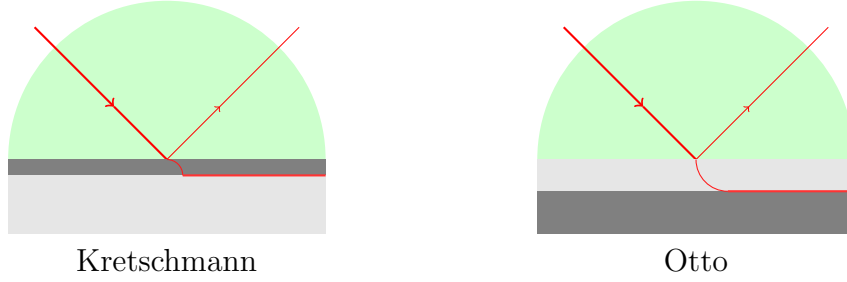


Figure 2.4 Prism coupling methods. Prism coupling to SPPs using attenuated total internal reflection. The Kretschmann configuration (left) is arranged as: prism-metal-dielectric and the Otto configuration (right) is arranged as: prism-dielectric-metal.

in which we use the fact that the spatial and temporal derivatives are interchangeable. Next, for linear media, optical properties can be described by the relative permittivity ε giving $\mathbf{D} = \varepsilon\varepsilon_0\mathbf{E}$. Considering the fact that $\varepsilon_0\mu_0 = 1/c^2$, then (II.12) is

$$\nabla \times \nabla \times \mathbf{E} = -\frac{\varepsilon}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} \quad (\text{II.13})$$

Since we have $0 = \nabla \cdot \mathbf{D} = \varepsilon_0 \nabla \cdot (\varepsilon \mathbf{E}) = \varepsilon_0 \mathbf{E} \cdot \nabla \varepsilon + \varepsilon_0 \varepsilon \nabla \cdot \mathbf{E}$, where the term $\nabla \varepsilon$ is negligible when the spatial variance of ε is small over one optical wavelength, we have in general

$$\nabla^2 \mathbf{E} - \frac{\varepsilon}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} = 0 \quad (\text{II.14})$$

Since the time dependence of the eigensolution is $e^{-i\omega t}$, we write $\mathbf{E}(x, y, z, t) = \mathbf{E}(x, y, z)e^{-i\omega t}$ to obtain the Helmholtz equation

$$(\nabla^2 + k_0^2 \varepsilon) \mathbf{E}(x, y, z) = 0 \quad (\text{II.15})$$

where $k_0 \equiv \omega/c$ is the wavevector of the light with frequency ω in vacuum. Next, as depicted in Fig. 2.2, x is the direction of propagation, z points in the direction normal to the metal surface and y is the metal surface transverse extension such that \mathbf{E} has no dependence on y , i.e. $\partial_y \mathbf{E} = 0$. The electric field can be written as $\mathbf{E}(x, y, z) = \mathbf{E}(z)e^{i\beta x}$ where β is the propagation constant. Along the transverse direction, the form of the eigensolution satisfies $\partial_z \mathbf{E}(z) = -ik_z \mathbf{E}(z)$. Since $\nabla^2 = \partial_x^2 + \partial_y^2 + \partial_z^2$, so it holds that

$$k_z^2 + \beta^2 = k_0^2 \varepsilon \quad (\text{II.16})$$

Here we summarize the eigenvalues of spatial derivatives:

$$\nabla^2 \mathbf{E} = -k_0^2 \varepsilon \mathbf{E}$$

$$\partial_x \mathbf{E} = i\beta \mathbf{E}$$

$$\partial_y \mathbf{E} = 0$$

$$\partial_z \mathbf{E} = ik_z \mathbf{E}$$

Now the dielectric function $\varepsilon = \varepsilon(z, \omega)$ is written explicitly as a piecewise constant function with a discontinuity at the dielectric-metal interface $z = 0$,

$$\varepsilon(z, \omega) = \begin{cases} \varepsilon_d(\omega) & \text{when } z \geq 0 \\ \varepsilon_m(\omega) & \text{when } z < 0 \end{cases} \quad (\text{II.17})$$

where subscripts d and m denote dielectric and metal respectively. The wavevector along the direction of propagation must be continuous across the interface, so β

remains the same for both media. The z component of the wavevector, however, differs in the two media

$$k_z(z) = \begin{cases} k_{dz} & \text{when } z \geq 0 \\ k_{mz} & \text{when } z < 0 \end{cases} \quad (\text{II.18})$$

The electric field can now be written in the form

$$\mathbf{E}(x, y, z, t) = \mathbf{E}_0 e^{i\beta x} e^{ik_z z} e^{-i\omega t} \quad (\text{II.19})$$

The physical attributes of the SPPs are clearly represented by the wavevectors (β and k_z), and the wavevectors can be determined in terms of the EM wave frequency ω via the medium's optical response - the dielectric function. Since the dielectric function $\varepsilon(\omega)$ is a function of wave frequency too, essentially we are looking for the dispersion relation of the eigensolution in the form of $k = k(\omega)$. To get the SPP dispersion via a relatively straightforward path, we step back to the general source-free Maxwell equations and then consider the boundary conditions with a specific medium structure.

The Maxwell equations in vector form are

$$\nabla \cdot \mathbf{D} = 0 \quad (\text{II.20a})$$

$$\nabla \cdot \mathbf{B} = 0 \quad (\text{II.20b})$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (\text{II.20c})$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} \quad (\text{II.20d})$$

The first two divergence equations provide the boundary conditions which will be used shortly and the last two curl equations provide the relations between different field components. To implement the SPP geometry layout, the last two equations can be rewritten separately by their Cartesian components in frequency ω domain

$$i\omega B_x = \partial_y E_z - \partial_z E_y \quad (\text{II.21a})$$

$$i\omega B_y = \partial_z E_x - \partial_x E_z \quad (\text{II.21b})$$

$$i\omega B_z = \partial_x E_y - \partial_y E_x \quad (\text{II.21c})$$

$$-i\omega D_x = \partial_y H_z - \partial_z H_y \quad (\text{II.21d})$$

$$-i\omega D_y = \partial_z H_x - \partial_x H_z \quad (\text{II.21e})$$

$$-i\omega D_z = \partial_x H_y - \partial_y H_x \quad (\text{II.21f})$$

In order to separate the 12 field components into two separate mode groups: we distinguish transverse electric (TE) modes and transverse magnetic (TM) modes.

The SPP supporting media have linear susceptibilities with scalar response so that $\mathbf{D} \parallel \mathbf{E}$ and $\mathbf{H} \parallel \mathbf{B}$. Further, the linearity dictates that the Cartesian components in the electric displacement \mathbf{D} field are solely determined by the corresponding electric \mathbf{E} field components, as indicated by the out-going arrows (black) in Fig. 2.5. Similarly the Cartesian components in the \mathbf{H} field are solely determined by the corresponding \mathbf{B} components, as indicated by the in-going arrow (black) in Fig. 2.5. Formally speaking, this linear dependency between the field pairs $\mathbf{D} - \mathbf{E}$ and $\mathbf{H} - \mathbf{B}$ need not be assumed. In such instances, the dielectric

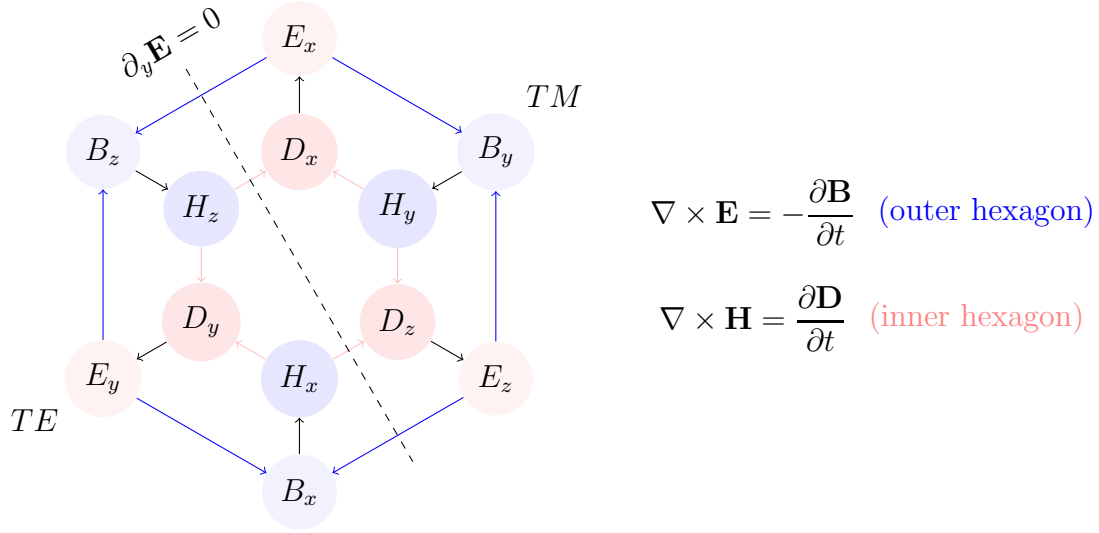


Figure 2.5 TE and TM mode separation. The outer hexagon shows that every B component is determined by two E components (blue arrows) and the inner hexagon shows that every D component is determined by two H components (pink arrows). The out-going and in-going arrows (black) indicate linear relations in $D - E$ and $H - B$ field pairs. The independence of all the fields in the y direction separate the field components into two mode sets: the Transverse Electric (TE) mode and the Transverse Magnetic (TM) mode with x the direction of propagation.

permittivity takes a diagonal tensor form. For this study, however, the chosen dielectric and the metal are isotropic media, so a scalar ε suffices.

For our SPP geometry choice, a geometrical constraint of the system is that the derivatives of the fields respect to y are zero. Since every component of the magnetic field, \mathbf{B} , is determined by two components in the electric field \mathbf{E} , and similarly every component of the electric displacement field \mathbf{D} is determined by two components

of the auxiliary magnetic field \mathbf{H} , via partial derivatives, as indicated by the outer arrows (blue) and inner arrows (pink), the condition $\partial_y \mathbf{E} = 0$ essentially detaches the $x \leftrightarrow z$ dependency in $E \rightarrow B$ and $H \rightarrow D$ as depicted by the dashed line in Fig. 2.5.

The eigenmode involving fields B_y, E_x and E_z is called Transverse Magnetic (TM) because the magnetic field B_y (or H_y) is in the transverse direction relative to the propagation direction \mathbf{x} , and the eigenmode involving fields E_y, H_x and H_z is called Transverse Electric (TE) for E_y transverse to \mathbf{x} :

$$\text{TM mode: } (H_y, E_x, E_z)$$

$$\text{TE mode: } (E_y, H_x, H_z)$$

The general boundary conditions from Maxwell's equations are

$$\mathbf{n} \cdot (\mathbf{D}_2 - \mathbf{D}_1) = \rho_e \quad (\text{II.22a})$$

$$\mathbf{n} \cdot (\mathbf{B}_2 - \mathbf{B}_1) = 0 \quad (\text{II.22b})$$

$$\mathbf{n} \times (\mathbf{E}_2 - \mathbf{E}_1) = 0 \quad (\text{II.22c})$$

$$\mathbf{n} \times (\mathbf{H}_2 - \mathbf{H}_1) = \mathbf{K}_e \quad (\text{II.22d})$$

where $\mathbf{n} = \hat{z}$ is the normal direction on the surface pointing from the metal into the dielectric.

Again, for the SPP configuration we consider, the surface charge density ρ_e and the surface current density \mathbf{K} are zero. The boundary conditions in terms of

Cartesian components at boundary $z = 0$ are:

$$D_{dz} - D_{mz} = 0 \quad (\text{II.23a})$$

$$B_{dz} - B_{mz} = 0 \quad (\text{II.23b})$$

$$\begin{cases} E_{dx} - E_{mx} = 0 \\ E_{dy} - E_{my} = 0 \end{cases} \quad (\text{II.23c})$$

$$\begin{cases} H_{dx} - H_{mx} = 0 \\ H_{dy} - H_{my} = 0 \end{cases} \quad (\text{II.23d})$$

here all fields marked with a subscript “d” are evaluated at $z = 0^+$ (the vacuum side) and all fields marked with subscript “m” are evaluated at $z = 0^-$ (the metal side).

Explicitly using (II.20) and (II.23) we derive that:

$$B_x : \quad k_{dz}E_y = k_{mz}E_y \quad (\text{II.24a})$$

$$B_y : \quad k_{dz}E_x - \beta E_{dz} = k_{mz}E_x - \beta E_{mz} \quad (\text{II.24b})$$

$$B_z : \quad \beta E_y = \beta E_y \quad (\text{II.24c})$$

$$D_x : \quad \frac{k_{dz}H_y}{\varepsilon_d} = \frac{k_{mz}H_y}{\varepsilon_m} \quad (\text{II.24d})$$

$$D_y : \quad \frac{k_{dz}H_x - \beta H_z}{\varepsilon_d} = \frac{k_{mz}H_x - \beta H_z}{\varepsilon_m} \quad (\text{II.24e})$$

$$D_z : \quad \beta H_y = \beta H_y \quad (\text{II.24f})$$

To satisfy (II.24) requires $E_y = 0$ (for example (II.24a)), so the eigenmode can only

be TM with H_z, E_x, E_y non-zero. The relation turns out:

$$\frac{k_{dz}}{\varepsilon_d} = \frac{k_{mz}}{\varepsilon_m} \quad (\text{II.25})$$

With the relation (II.16) $\beta^2 + k_z^2 = k_0^2\varepsilon$, the dispersion relation is derived:

$$\beta^2 = k_0^2 \frac{\varepsilon_d \varepsilon_m}{\varepsilon_d + \varepsilon_m} \quad (\text{II.26a})$$

$$k_{dz}^2 = k_0^2 \frac{\varepsilon_d^2}{\varepsilon_d + \varepsilon_m} \quad (\text{II.26b})$$

$$k_{mz}^2 = k_0^2 \frac{\varepsilon_m^2}{\varepsilon_d + \varepsilon_m} \quad (\text{II.26c})$$

We also note the following relations between the field components from (II.24b):

$$E_{dz} = -\frac{k_{mz}}{\beta} E_x \quad (\text{II.27a})$$

$$E_{mz} = -\frac{k_{dz}}{\beta} E_x \quad (\text{II.27b})$$

Using the relations above, we see that the SPP profile is completely determined by its frequency (or wavelength) and the optical response of the supporting dielectric-metal.

In the ideal case, the metal is non-absorptive and so the dielectric function is a real negative number. Thus the propagation factor β is real and positive while the transverse decay factors k_{dz} and k_{mz} are purely imaginary. In reality, due to the finite absorption in a metal, the dielectric function of the metal has a non-zero imaginary component, therefore all quantities are complex numbers. For a laser source with wavelength $\lambda = 800nm$, the corresponding SPP propagation length L_{spp} and penetration lengths in the dielectric δ_d , and in the metal δ_m can be easily

derived from the expressions of SPP wavevectors derived above. This is shown schematically in Fig. 2.6a for the following values:

$$L_{spp} = 1/\text{Im}[\beta] = 80\mu m$$

$$\delta_d = 1/\text{Im}[k_{dz}] = 635nm$$

$$\delta_m = 1/\text{Im}[k_{mz}] = 25nm$$

A real-time snapshot of the SPP field profile is shown in Fig. 2.6b, calculated using a Finite-difference time-domain (FDTD) simulation.

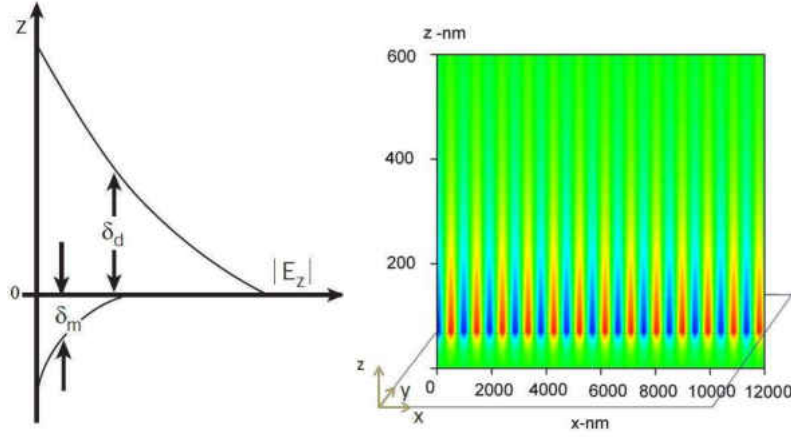


Figure 2.6 The penetration depths and a snapshot of the SPPs. (a) Penetration depth of an SPP in a dielectric and metal (Figure taken from ref. [10]) (b) The SPP instantaneous profile snapshot, represented by the magnetic $\mathbf{H} = \hat{y}H_y$ field only. This picture is achieved by an FDTD simulation written by the author, and the source code is attached in Appendix E.

CHAPTER III

INTRODUCTION TO NONLINEAR MEDIA AND NONLINEAR WAVE EQUATIONS

Introduction to Nonlinear Optics

This work was published in APS March Meeting in 2010. Dr. Miriam Deutsch set up the proposed frame of the work I was the primary contributor to all the derivations and analysis of the results.

Before the invention of the laser, light sources used in laboratories were of relatively low intensity, therefore most optical phenomena observed in labs were linear, and most experimental results could be explained simply by the superposition principles. In reality, however, The optical properties of a material system can be modified by the presence of high intensity light. This light-induced modification was first demonstrated in the second harmonic generation experiment by Franken *et al.* in 1961 [58]. This experiment, which occurred only one year after the demonstration of the first working laser in 1960, marked the birth of a new field of research in physics: nonlinear optics.

A lot of materials can show nonlinearity at different thresholds of light intensity. Even in vacuum photons can interact via vacuum polarizations and show

a certain level of nonlinearity in the context of wave mixing [59]. In practice, certain dielectric media are preferred such as glass and semiconductors [60] to observe nonlinear optical effects. This is because nonlinear effects are greatly enhanced by the medium polarization. Numerous new nonlinear optical phenomena have been discovered in vast types and states of materials [61] [62] [63] [64]. The microscopic origin of the nonlinearities in materials is classically illustrated in the nonlinear oscillator model [39] [40].

Metals, contrary to our intuition, possess very high values of intrinsic nonlinearity even compared to fiber glasses (SiO_2) [41]. It is rare to notice the metallic nonlinearities, however, due to the limited penetration depth of light into the bulk metals which is typically less than one tenth of a wavelength of the light wave. Therefore the nonlinear interactions between the light and metals are very weak because the volume of material available to interact with the light is small. The SPP waves, on the other hand, can concentrate high intensity of light onto the metal surface. The SPPs also greatly increase the interaction volume between the light and the metal by the guided nature of the SPP waves. Therefore, the SPPs are one of the best candidates to observe nonlinearity of the metals. There have been already different orders of metal nonlinearities demonstrated by harmonic generations [65] [66] and by four wave mixing [56] [67].

Wave Equations in a Nonlinear Medium

The nonlinearity of medium can be directly expressed by means of the macroscopic polarization \mathbf{P} , which appears in the definition of electric displacement \mathbf{D} , as shown in (II.3). Under the dipole approximation, which is the lowest order expression of \mathbf{P} which neglects the magnetic terms, the polarization \mathbf{P} can be expressed as an expansion of the electric field \mathbf{E} in polynomial orders. The relations between polarization $\mathbf{P}(\omega)$, electric displacement \mathbf{D} and electric field $\mathbf{E}(\omega)$ are

$$\mathbf{P}(\omega) = \varepsilon_0(\chi^{(1)}\mathbf{E} + \chi^{(2)}\mathbf{E}\mathbf{E} + \chi^{(3)}\mathbf{E}\mathbf{E}\mathbf{E} + \dots) \quad (\text{III.1a})$$

$$\mathbf{D}(\omega) = \varepsilon_0\mathbf{E} + \mathbf{P} = \varepsilon_0(\mathbf{E} + \chi^{(1)}\mathbf{E} + \chi^{(2)}\mathbf{E}\mathbf{E} + \chi^{(3)}\mathbf{E}\mathbf{E}\mathbf{E} + \dots) \quad (\text{III.1b})$$

where the $\chi^{(n)}(\omega)$ terms are the susceptibilities of different orders. The first two terms, \mathbf{E} and $\chi^{(1)}\mathbf{E}$ are recognized as the linear parts and can be merged and expressed by the dielectric function:

$$\varepsilon(\omega) = 1 + \chi^{(1)}(\omega) \quad (\text{III.2})$$

and the rest is named as nonlinear polarization:

$$\mathbf{P}^{NL}(\omega) = \varepsilon_0(\chi^{(2)}\mathbf{E}\mathbf{E} + \chi^{(3)}\mathbf{E}\mathbf{E}\mathbf{E} + \dots) \quad (\text{III.3})$$

Gather the equations above, together with (II.12), and the nonlinear wave equation in a nonlinear medium is achieved with the form:

$$\nabla \times \nabla \times \mathbf{E} + \frac{\varepsilon}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} = -\mu_0 \frac{\partial^2 \mathbf{P}^{NL}}{\partial t^2} \quad (\text{III.4})$$

In general, $\mathbf{P}^{NL} = \mathbf{P}^{NL}(\mathbf{E})$ is a function of electric field \mathbf{E} expanded with polynomial terms. In order to simplify and solve the nonlinear wave equation, the detailed nonlinear properties of the material in which the light propagates needs to be discussed and expressed in a specific form of a response function.

For noble metals such as silver and gold which possess inversion symmetry, the even orders of $\chi^{(n)}$ vanish and the dominating nonlinear term is then $\chi^{(3)}$ [39]. The nonlinear polarization is now simplified to

$$\mathbf{P}^{NL}(\omega) = \varepsilon_0 \chi^{(3)}(\omega : \omega_\alpha, \omega_\beta, \omega_\gamma) \mathbf{E}(\omega_\alpha) \mathbf{E}(\omega_\beta) \mathbf{E}(\omega_\gamma) \quad (\text{III.5})$$

where subscripts α, β and γ denote the input electric field distinguishable by their angular frequencies ω or input momentum vectors \mathbf{k} , and the resultant polarization frequency is a linear combination of $\omega_\alpha, \omega_\beta$ and ω_γ , for the energy is conserved. This is generally known as a parametric process.

In our study, we use degenerate input with the same frequency, i.e. $\omega_a = \omega_b = \omega_c = \omega$. Among several possibilities, there are two possible resultant frequencies: $3\omega (= \omega + \omega + \omega)$ Fig. 3.1 and $\omega (= \omega + \omega - \omega)$ Fig. 3.2. The first case with 3ω is introduced as tripled frequency in section .1 and discussed in details in Chapter IV as third harmonic generation. The second case with ω is introduced as nonlinear optical index in section .2 and the details of the resultant soliton solution are discussed in Chapter V.

Sum Frequency - Harmonic Generation

One of the resultant nonlinear polarizations has a frequency that comprises

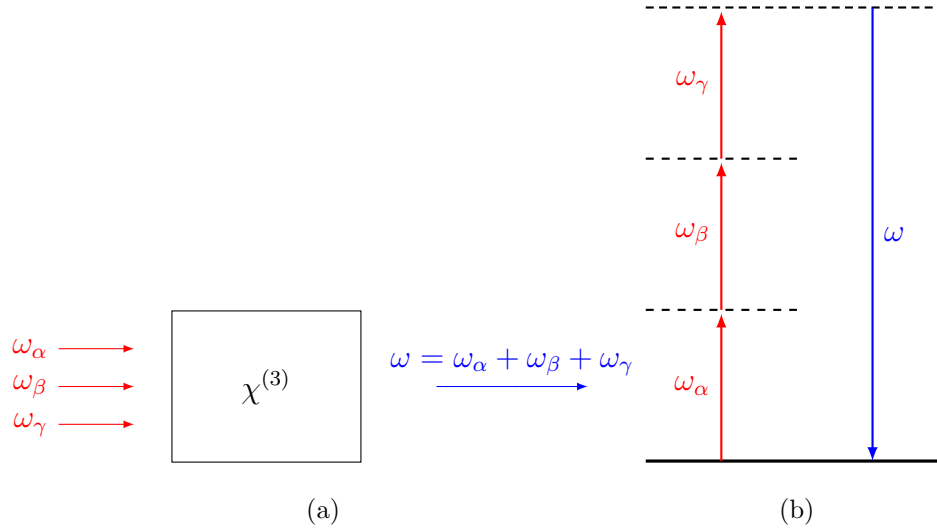


Figure 3.1 Sum frequency generation. (a) Sum frequency generation via nonlinear material. Three incident beams with frequencies ω_α , ω_β and ω_γ interact within the nonlinear medium with third order susceptibility $\chi^{(3)}$. A resultant sum frequency signal $\omega = \omega_\alpha + \omega_\beta + \omega_\gamma$ is generated. (b) Schematic illustration of energy levels for the sum frequency process. Three photons at lower frequency are destroyed and a photon is generated at a the frequency of sum of the destroyed photons.

the summation of the frequencies of the input light waves, known as the sum frequency process, where three photons are converted into a new photon with much higher energy (or frequency) Fig. 3.1b. For a sum frequency process with the same degenerate frequency $3\omega = \omega + \omega + \omega$, the nonlinear polarization term is now:

$$P_i(3\omega) = \varepsilon \sum_{jkl} \chi_{ijkl}^{(3)}(3\omega) E_j(\omega) E_k(\omega) E_l(\omega) \quad (\text{III.6})$$

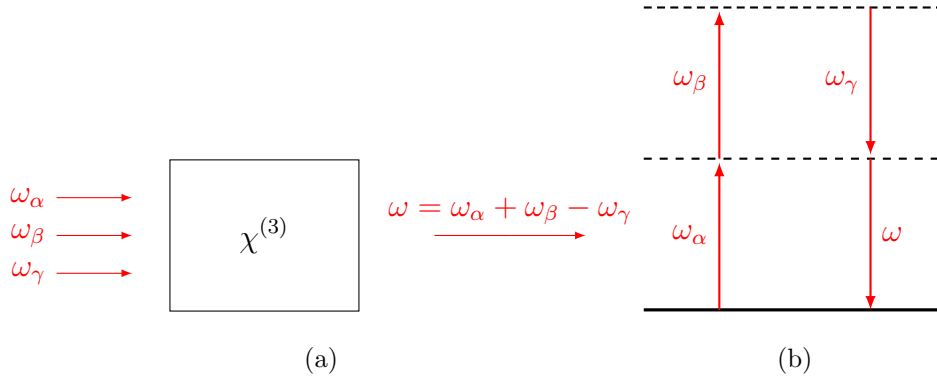


Figure 3.2 Difference frequency generation. (a) Difference frequency generation via nonlinear material. Three incident beams with frequencies ω_α , ω_β and ω_γ interact within the nonlinear medium with third order susceptibility $\chi^{(3)}$, a resultant difference frequency signal $\omega = \omega_\alpha + \omega_\beta - \omega_\gamma$ is generated. (b) Schematic illustration of energy levels for the difference frequency process.

and this special case of the general sum frequency processes is also called third harmonic generation.

Now the electric field \mathbf{E} in (III.4) has two frequency components and may be written in the form:

$$\mathbf{E} = \mathbf{E}(\omega)e^{-i\omega t} + \mathbf{E}(3\omega)e^{-3i\omega t} \quad (\text{III.7})$$

Therefore the nonlinear equation (III.4) is Fourier transformed into the frequency domain by $t \rightarrow \omega$:

$$\begin{aligned} \nabla \times \nabla \times [\mathbf{E}(\omega)e^{-i\omega t} + \mathbf{E}(3\omega)e^{-3i\omega t}] - \frac{\varepsilon}{c^2}[\omega^2\mathbf{E}(\omega)e^{-i\omega t} + (3\omega)^2\mathbf{E}(3\omega)e^{-3i\omega t}] \\ = \mu_0[\omega^2\mathbf{P}^{NL}(\omega)e^{-i\omega t} + (3\omega)^2\mathbf{P}^{NL}(3\omega)e^{-3i\omega t}] \end{aligned} \quad (\text{III.8})$$

It is obvious that the electric and polarization fields dependent on ω and

3ω are coupled via (III.6) and (III.8), but the two equations can not be solved simultaneously with exact analytical methods. In order to further simplify the equations, the fields with the same frequency dependence are grouped together and separated from other frequency components. Therefore the non-depletion (ND) approximation is used to decouple the equations with different frequencies. The ND approximation states that the nonlinear polarization drains very little energy from the input source and that essentially the change of the input field is negligible. The ND approximation is valid only for high input with low efficiency output. The ND approximation insures the nonlinear polarization is solely determined by (III.6) and will not affect the input electric field $\mathbf{E}(\omega)$. This essentially eliminates the dependence of between the 3ω and ω components of the fields in (III.6). Now (III.6) can be written as two separate equations by matching the oscillation frequencies. The one with 3ω frequency is

$$\nabla \times \nabla \times \mathbf{E}(3\omega) - \frac{\varepsilon}{c^2}(3\omega)^2 \mathbf{E}(3\omega) = \mu_0(3\omega)^2 \mathbf{P}^{NL}(3\omega) \quad (\text{III.9})$$

After the nonlinear polarization $\mathbf{P}^{NL}(3\omega)$ is determined from the input fields $\mathbf{E}(\omega)$, the generated harmonic fields $\mathbf{E}(3\omega)$ are solved from (III.9). The incident linear SPP wave preserves its longitudinal and transverse profiles within the ND approximation, thus we are looking for a steady-state solution for the output field in the form $\mathbf{E}(\mathbf{r}, t) = \mathbf{E}(\mathbf{r})e^{\omega_s t}$ with a steady-state solution frequency $\omega_s = 3\omega_0$. We ignore the transient solutions. The validity and standard procedure of the steady-state solution approach are discussed in Appendix A.

Difference Frequency - Nonlinear Optical Index

The optical refractive index of many materials depends on the intensity of the incident light and can be described by the relation

$$n = n_0 + n_2|E|^2 \quad (\text{III.10})$$

where n_0 is the linear refractive index and n_2 is the nonlinear index of refraction [41]. The phenomena of a light intensity dependent optical index is sometimes referred to as the optical Kerr effect. The nonlinear polarization in the nonlinear Kerr effect is determined by

$$\mathbf{P}^{NL}(\omega) = 3\chi^{(3)}(\omega = \omega + \omega - \omega)|E(\omega)|^2\mathbf{E}(\omega) \quad (\text{III.11})$$

where the intensity $I \sim |E|^2$.

The expression of the nonlinear polarization is a special case of (III.5) in difference frequency generation Fig. 3.2 when the frequencies of the incident light degenerate $\omega_\alpha = \omega_\beta = \omega_\gamma = \omega$. Since the resultant nonlinear polarization has the same frequency of the incident electric fields, the ND approximation for the frequency decoupling method from last section is not applicable, no matter how small the nonlinear polarization. The induced and original electric fields are indistinguishable and the original beam is treated as it undergoes a nonlinear refractive index. A new approach is needed to solve the nonlinear wave equation and we turn to some other approximations and techniques, known as the Nonlinear

Schödinger (NLS) Equation, which will be introduced and discussed in detail in Chapter V.

Simplifications of the Nonlinearities in an SPP System

In the previous sections, the temporal property of the third order susceptibility $\chi^{(3)}$ was separated into two different frequency cases - sum and difference frequency processes. In this section we discuss the spatial attributes of the third order susceptibility $\chi^{(3)}$. This discussion will be based on the sum frequency process, but the same theory can be applied to the difference frequency process straightforwardly.

The third order susceptibility $\chi^{(3)}$ has a general form of a rank 4 tensor with 81 components. For example, the sum frequency polarization component in Cartesian coordinates is written:

$$\begin{aligned}
 P_i(\omega : \omega_a, \omega_b, \omega_c) &= \varepsilon_0 \sum_{jkl} \sum_{(abc)} \chi_{ijkl}^{(3)}(\omega : \omega_a, \omega_b, \omega_c) E_j(\omega_a) E_k(\omega_b) E_l(\omega_c) \\
 &= \varepsilon_0 g \sum_{jkl} \chi_{ijkl}^{(3)}(\omega) E_j^a E_k^b E_l^c
 \end{aligned} \tag{III.12}$$

The subscripts i, j, k, l represent the permutations of Cartesian axis x, y and z , with the distinguishable input fields labeled with a, b and c . The g factor is the degeneracy factor for all different frequency combinations and contains all the information about frequency mixing discussed in previous sections. For example in the degenerate case $\omega_\alpha = \omega_\beta = \omega_\gamma = \omega$, $g(3\omega) = 1$ since there is only one way to

produce the third harmonic generation

$$3\omega = \omega + \omega + \omega \quad (\text{III.13})$$

while $g(\omega) = 3$ since there are three ways to produce the same frequency of ω :

$$\begin{aligned} \omega &= \omega + \omega - \omega \\ \omega &= \omega - \omega + \omega \\ \omega &= -\omega + \omega + \omega \end{aligned} \quad (\text{III.14})$$

There are many independent tensor elements in $\chi^{(3)}$. Fortunately, many of these tensor elements can be eliminated by symmetry and proper choice of the coordinate system. Since the crystal structures of gold and silver are both face centered cubic (FCC), the independent elements in the tensor is reduced to 21 nonzero elements with only 4 of them independent [68] [69]: $\chi_{1111}, \chi_{1122}, \chi_{1212}, \chi_{1221}$, where the subscripts 1 and 2 stand for any two orthogonal Cartesian axis parallel to the optical axis of the crystal lattice, for example $\chi_{xxyy} = \chi_{yyzz} = \chi_{zzyy} = \chi_{zzxx} = \dots = \chi_{1122}$.

There are three different coordinate systems in which the electric field is represented: the coordinates attached to the optical axis of the cubic crystal lattice (*cubic*, labeled by Greek symbols α, β and γ), the orientation of the crystal lattice (*crystal*, labeled by letters with a prime '), and the lab coordinates (*lab*, labeled by letters) where actual experimental measurement occurs. A simple illustration of the coordinate systems in 2-dimension is presented in Fig. 3.3.

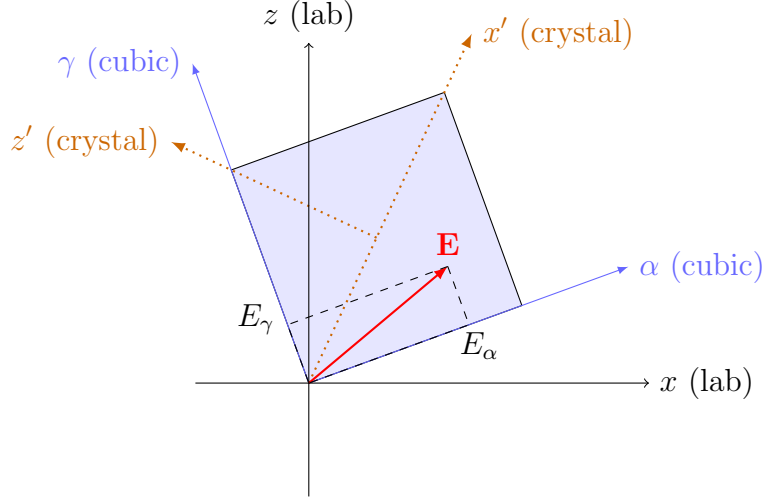


Figure 3.3 Three coordinate systems are illustrated here. The components of the electric field \mathbf{E} are represented relative the optical axis (*cubic*). For each coordinate system, only two axes are shown for clarity.

The nonlinear polarization \mathbf{P}^{NL} can be written in component form, in both cubic and lab Cartesian coordinate systems:

$$\mathbf{P}^{NL} = \hat{\alpha}P_{\alpha} + \hat{\beta}P_{\beta} + \hat{\gamma}P_{\gamma} = \hat{x}P_x + \hat{y}P_y + \hat{z}P_z \quad (\text{III.15})$$

The nonlinear polarization \mathbf{P}^{NL} and the electric field \mathbf{E} have straightforward relations in the cubic coordinate system:

$$\begin{aligned} P_{\alpha}^{NL} = & \chi_{1111}E_{\alpha}^a E_{\alpha}^b E_{\alpha}^c \\ & + \chi_{1122}E_{\alpha}^a (\mathbf{E}^b \cdot \mathbf{E}^c) - \chi_{1122}E_{\alpha}^a E_{\alpha}^b E_{\alpha}^c \\ & + \chi_{1212}E_{\alpha}^b (\mathbf{E}^c \cdot \mathbf{E}^a) - \chi_{1212}E_{\alpha}^a E_{\alpha}^b E_{\alpha}^c \\ & + \chi_{1221}E_{\alpha}^c (\mathbf{E}^a \cdot \mathbf{E}^b) - \chi_{1221}E_{\alpha}^a E_{\alpha}^b E_{\alpha}^c \end{aligned} \quad (\text{III.16a})$$

$$\begin{aligned}
P_\beta^{NL} &= \chi_{1111} E_\beta^a E_\beta^b E_\beta^c \\
&+ \chi_{1122} E_\beta^a (\mathbf{E}^b \cdot \mathbf{E}^c) - \chi_{1122} E_\beta^a E_\beta^b E_\beta^c \\
&+ \chi_{1212} E_\beta^b (\mathbf{E}^c \cdot \mathbf{E}^a) - \chi_{1212} E_\beta^a E_\beta^b E_\beta^c \\
&+ \chi_{1221} E_\beta^c (\mathbf{E}^a \cdot \mathbf{E}^b) - \chi_{1221} E_\beta^a E_\beta^b E_\beta^c
\end{aligned} \tag{III.16b}$$

$$\begin{aligned}
P_\gamma^{NL} &= \chi_{1111} E_\gamma^a E_\gamma^b E_\gamma^c \\
&+ \chi_{1122} E_\gamma^a (\mathbf{E}^b \cdot \mathbf{E}^c) - \chi_{1122} E_\gamma^a E_\gamma^b E_\gamma^c \\
&+ \chi_{1212} E_\gamma^b (\mathbf{E}^c \cdot \mathbf{E}^a) - \chi_{1212} E_\gamma^a E_\gamma^b E_\gamma^c \\
&+ \chi_{1221} E_\gamma^c (\mathbf{E}^a \cdot \mathbf{E}^b) - \chi_{1221} E_\gamma^a E_\gamma^b E_\gamma^c
\end{aligned} \tag{III.16c}$$

Together, in cubic coordinates (in terms of α, β, γ)

$$\begin{aligned}
\mathbf{P}^{NL} &= \chi_{1122} \mathbf{E}^a (\mathbf{E}^b \cdot \mathbf{E}^c) + \chi_{1212} \mathbf{E}^b (\mathbf{E}^c \cdot \mathbf{E}^a) + \chi_{1221} \mathbf{E}^c (\mathbf{E}^a \cdot \mathbf{E}^b) \\
&+ \chi_d (\hat{\alpha} E_\alpha^a E_\alpha^b E_\alpha^c + \hat{\beta} E_\beta^a E_\beta^b E_\beta^c + \hat{\gamma} E_\gamma^a E_\gamma^b E_\gamma^c)
\end{aligned} \tag{III.17}$$

where the first three terms are invariant under crystal rotations, while the last term with $\chi_d \equiv \chi_{1111} - \chi_{1122} - \chi_{1212} - \chi_{1221}$ shows anisotropy under crystal rotations. Expressing the polarization in a compact form under lab coordinate yields:

$$\mathbf{P}^{NL} = \chi_{1122} (\mathbf{E}^b \cdot \mathbf{E}^c) \mathbf{E}^a + \chi_{1212} (\mathbf{E}^c \cdot \mathbf{E}^a) \mathbf{E}^b + \chi_{1221} (\mathbf{E}^a \cdot \mathbf{E}^b) \mathbf{E}^c + \chi_d M :: \mathbf{E}^a \mathbf{E}^b \mathbf{E}^c \tag{III.18}$$

Therefore the tensor M , also known as the rotation matrix, takes different values for different crystal orientations and should be considered in separate cases which are detailed in Appendix C. The simplest case, which we apply to our nonlinear SPP structure is oriented along the (100) crystal orientation

with no rotation. In this simplifying case, together with the degenerate input approximation, the nonlinear polarization (III.18) is simplified to

$$\chi_d M :: \mathbf{E}^a \mathbf{E}^b \mathbf{E}^c = \chi_d (\hat{\mathbf{x}} E_x^a E_x^b E_x^c + \hat{\mathbf{y}} E_y^a E_y^b E_y^c + \hat{\mathbf{z}} E_z^a E_z^b E_z^c) \quad (\text{III.19})$$

In the following chapters we are going to use this simplest configuration to illustrate nonlinear SPP properties without the cumbersome geometry distractions.

CHAPTER IV

HARMONIC GENERATION IN SURFACE PLASMON POLARITONS

Theoretical Development of Third Harmonic Generation of Nonlinear SPPs

This work was published in volume 696 of *Frontiers in Optics (FiO)/Laser Science XXVI (LS)* conference in 2010. Dr. Miriam Deutsch set up the proposed frame of the work I was the primary contributor to all the derivations and analysis of the results.

In this chapter, we theoretically demonstrate that it is possible to utilize the effect of third order susceptibility $\chi^{(3)}$ in noble metals to induce third harmonic generation (THG) from an SPP wave. The research on second harmonic generation (SHG) on the metal surface by SPPs can be traced back more than two decades ago on and it has been explored on a variety of metals, e.g. gold, silver and aluminum [70] [71], as well as on different geometric configurations, e.g. bulk, grating and film [71] [72] [73], all based on second order susceptibility $\chi^{(2)}$. The effect of the intrinsic third order susceptibility $\chi^{(3)}$ of metals, however, were not covered until recently by Novotny and et. al. [56] [67] in the SPP four wave mixing experiments.

We start our study under the non-depletion approximation as stated in

previous chapter. The wave equation (III.4) is

$$\nabla \times \nabla \times \mathbf{E} + \frac{\varepsilon}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} = -\mu_0 \frac{\partial^2 \mathbf{P}^{NL}}{\partial t^2} \quad (\text{IV.1})$$

The steady-state solution of this wave equation has the form

$$\mathbf{E}(\mathbf{r}, t) = \mathbf{E}(\mathbf{r}) e^{-i\omega_s t} \quad (\text{IV.2})$$

The spatial part of the field, $\mathbf{E}(\mathbf{r})$, can be written as the sum of the transverse and longitudinal components

$$\mathbf{E} = \mathbf{E}_\perp + \mathbf{E}_\parallel \quad (\text{IV.3})$$

where the transverse component relative to the propagation wavevector is labeled by \perp and the longitudinal component by \parallel . Using the Helmholtz theorem in Appendix B, the equation can be transformed according to:

$$\nabla \times \nabla \times \mathbf{E}(\mathbf{r}) \Rightarrow -(ik)^2 \mathbf{E}_\perp(\mathbf{k}) \quad (\text{IV.4})$$

while the temporal part, by Fourier transformation into frequency domain, is given by

$$\frac{\partial^2}{\partial t^2} \mathbf{E}(\mathbf{r}) \Rightarrow -\omega_s^2 \mathbf{E}(\omega) \quad (\text{IV.5})$$

where ω_s represents steady-state solution frequency. Now the nonlinear wave equation(III.4) can be written as

$$-(ik)^2 \mathbf{E}_\perp(\omega, \mathbf{k}) - \varepsilon \frac{\omega_s^2}{c^2} \mathbf{E}(\omega, \mathbf{k}) = \mu_0 \omega_s^2 \mathbf{P}^{NL}(\omega_s, \mathbf{k}) \quad (\text{IV.6})$$

The transverse \perp and longitudinal \parallel components of the equation can be achieved by respectively applying $\mathbf{k}\times$ and $\mathbf{k}\cdot$ on (IV.6) which produces the two separate equations:

$$-(i\mathbf{k})^2\mathbf{E}_\perp - \varepsilon\frac{\omega_s^2}{c^2}\mathbf{E}_\perp = \mu_0\omega_s^2\mathbf{P}_\perp^{NL} \quad (\text{IV.7a})$$

$$-\varepsilon\varepsilon_0\mathbf{k}\cdot\mathbf{E}_\parallel = \mathbf{k}\cdot\mathbf{P}_\parallel^{NL} \quad (\text{IV.7b})$$

Performing an inverse Fourier transform on the equations above gives the equations in component form [40]:

$$\nabla^2\mathbf{E}_\perp(\mathbf{r},\omega) + \varepsilon\frac{\omega_s^2}{c^2}\mathbf{E}_\perp = -\mu_0\omega_s^2\mathbf{P}_\perp^{NL} \quad (\text{IV.8a})$$

$$\nabla\cdot[\varepsilon\varepsilon_0\mathbf{E}_\parallel + \mathbf{P}_\parallel^{NL}] = 0 \quad (\text{IV.8b})$$

These are the central equations we are going to focus on.

To solve (IV.8a) and (IV.8b), first the explicit expression of \mathbf{P}^{NL} is needed. Continuing from the discussion in last chapter, here a cubic medium (noble metal) in (100) orientation with $\varphi = 0$ rotation is constructed for the geometry setup, so that (III.18)

$$\mathbf{P}^{NL} = \chi_{1122}(\mathbf{E}^b\cdot\mathbf{E}^c)\mathbf{E}^a + \chi_{1212}(\mathbf{E}^c\cdot\mathbf{E}^a)\mathbf{E}^b + \chi_{1221}(\mathbf{E}^a\cdot\mathbf{E}^b)\mathbf{E}^c + \chi_d M :: \mathbf{E}^a\mathbf{E}^b\mathbf{E}^c \quad (\text{IV.9})$$

where $\chi_d = \chi_{1111}^{(3)} - \chi_{1122}^{(3)} - \chi_{1212}^{(3)} - \chi_{1221}^{(3)}$, is now simplified to

$$P_i = 3\chi_{1212}^{(3)}E_i(\mathbf{E}\cdot\mathbf{E}) + (\chi_{1111}^{(3)} - 3\chi_{1212}^{(3)})E_i^3 \quad (\text{IV.10})$$

where $i = x, y, z$ is the Cartesian component index.

We define two constant factors $A \equiv \chi_{1111}^{(3)}$ and $B \equiv 3\chi_{1212}^{(3)}$, and only include terms of third harmonic generation:

$$P_i = BE_i|E|^2 + (A - B)E_i^3 \quad (\text{IV.11})$$

The incident SPP wave form is predetermined as

$$\mathbf{E} = (\hat{\mathbf{x}}E_{x0} + \hat{\mathbf{z}}E_{z0})e^{-i\omega_1 t + i\mathbf{k}_1 \cdot \mathbf{r}} \quad (\text{IV.12})$$

The subscript 0 denotes amplitude value of the field at the interface, ω_1 is the carrier frequency and $\mathbf{k}_1 = \mathbf{k}(\omega_1)$ is the propagation wavevector of the SPP. Consequently the components of \mathbf{P}^{NL} are written explicitly as

$$P_x^{NL} = BE_x|E|^2 + (A - B)E_x^3 = E_{x0}(A|E_{x0}|^2 + B|E_{z0}|^2)e^{-i\omega_3 t + i\mathbf{q} \cdot \mathbf{r}} \quad (\text{IV.13a})$$

$$P_y^{NL} = 0 \quad (\text{IV.13b})$$

$$P_z^{NL} = BE_z|E|^2 + (A - B)E_z^3 = E_{z0}(A|E_{z0}|^2 + B|E_{x0}|^2)e^{-i\omega_3 t + i\mathbf{q} \cdot \mathbf{r}} \quad (\text{IV.13c})$$

We the $\omega_3 = 3\omega_1$ frequency, the triple wavevector is defined $\mathbf{q} \equiv 3\mathbf{k}_1$, as well as the wavevector at triple frequency $\mathbf{K} \equiv \mathbf{k}(\omega_3)$. Since the nonlinear polarization has the dependency factor $e^{-i\omega_3 t + i\mathbf{q} \cdot \mathbf{r}}$, the steady-state frequency is determined by $\omega_s = \omega_3 = 3\omega_1$, and the solution to (IV.8b) and (IV.8a) can be solved separately in the dielectric and metal. Using the method of undetermined coefficients, and considering the fact that $\mathbf{P}^{NL} = 0$ in the linear dielectric, the solutions take the

forms:

$$\text{in dielectric: } \mathbf{E}_d = \mathbf{E}_{d0} e^{i\mathbf{K}_d \cdot \mathbf{r}} \quad (\text{IV.14a})$$

$$\text{in metal: } \mathbf{E}_m = \mathbf{E}_{m0} e^{i\mathbf{K}_m \cdot \mathbf{r}} + \mathbf{Q}_0 e^{i\mathbf{q} \cdot \mathbf{r}} \quad (\text{IV.14b})$$

The subscript 0 denotes that the amplitude values are taken on the interface between the dielectric d and the metal m . Here the \mathbf{E}_{m0} term is a free wave term that represents a radiating electro-magnetic field from the interface, while \mathbf{Q}_0 term is a driven term that represents an oscillating field acting as the secondary source to the generated radiating wave. The interface amplitudes \mathbf{E}_{d0} and \mathbf{E}_{m0} , as well as the wavevectors \mathbf{K}_d , \mathbf{K}_m , which depend on the tripled frequency ω_3 , are determined by boundary conditions. The interface amplitude $\mathbf{Q}_0 = \mathbf{Q}_\perp + \mathbf{Q}_\parallel$, on the other hand, is solved directly from the nonlinear polarization \mathbf{P}^{NL} . By solving (IV.8a) and (IV.8b) we get

$$\mathbf{Q}_\perp = \frac{\mu_0 \omega_3^2 \mathbf{P}_\perp^{NL}}{K_m^2 - q^2} \quad (\text{IV.15a})$$

$$\mathbf{Q}_\parallel = -\frac{\mathbf{P}_\parallel^{NL}}{\varepsilon_m \varepsilon_0} \quad (\text{IV.15b})$$

thus $\mathbf{Q}_0 = \mathbf{Q}_\perp + \mathbf{Q}_\parallel$ is determined. The longitudinal and transverse components of a field are defined relative to the wavevector \mathbf{k} according to Helmholtz theorem in Appendix A. For example the polarization has the components defined as

$$\mathbf{P}_\parallel = \frac{\mathbf{P} \cdot \mathbf{k}^*}{|k|^2} \mathbf{k} \quad (\text{IV.16a})$$

$$\mathbf{P}_\perp = \frac{\mathbf{k}^* \times (\mathbf{P} \times \mathbf{k})}{|k|^2} = \mathbf{P} - \mathbf{P}_\parallel \quad (\text{IV.16b})$$

For the next step, we need to determine all the wavevectors \mathbf{K}_d , \mathbf{K}_m and \mathbf{q} . As stated above, \mathbf{q} is already known and solely determined by form of the nonlinear polarization \mathbf{P}^{NL} so that $\mathbf{q} = 3\mathbf{k}_1$ where $\mathbf{k}_1 = \mathbf{k}_{spp}(\omega_1)$ is the incident SPP propagation wavevector at frequency ω_1 . Since we are looking for a steady-state solution, all the wavevectors have to match along the x direction, so that

$$K_{dx} = K_{mx} = q_x \quad (\text{IV.17})$$

With the relations that in the media $K_{dx}^2 + K_{dz}^2 = \varepsilon_d(\omega_3)\frac{\omega_3^2}{c^2}$ and $K_{mx}^2 + K_{mz}^2 = \varepsilon_m(\omega_3)\frac{\omega_3^2}{c^2}$, the z components of the wavevectors are now determined too. Thus all the wavevectors in the 3ω frequency regime are determined at this point. The imaginary part in K_d dominates, which indicates that the THG on the dielectric side is still bound onto the surface, shown in Fig. 4.1. In the metal, the real part in K_m dominates, which indicates that the THG in the metal is able to radiate to distance, shown in Fig. 4.2.

Next, to achieve all the field amplitudes, it is convenient to define the wavevector angle θ by wavevector components:

$$\tan \theta \equiv \frac{k_z}{k_x} \quad (\text{IV.18})$$

Also for the electric fields \mathbf{E}_{d0} and \mathbf{E}_{m0} , it is convenient to define an assistant angle φ that $\varphi = \theta + \frac{\pi}{2}$ because $\mathbf{k} \cdot \mathbf{E} = 0$ and it satisfies

$$\tan \varphi = \frac{E_z}{E_x} \quad (\text{IV.19})$$

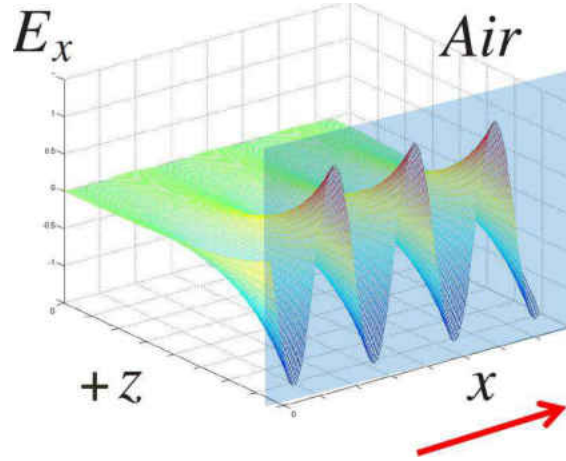


Figure 4.1 A graph of the THG solution in vacuum. The THG solution in vacuum is still bounded on the surface. Here the E-field component E_x is used in the graph.

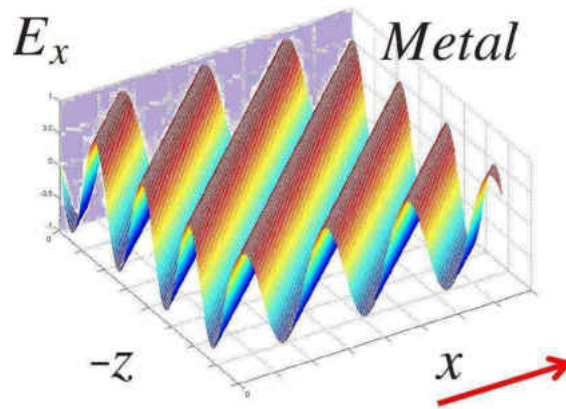


Figure 4.2 A graph of the THG solution in metal. The THG solution in metal is a propagating wave. The propagation angle is determined by the wavevector \mathbf{K}_d . Here the E-field component E_x is used in the graph.

Now we can find out all the field amplitudes on the interface by matching the boundary conditions: The boundary condition for E along x direction at $z = 0$ is

$$\hat{\mathbf{x}} \cdot \mathbf{E}_d = \hat{\mathbf{x}} \cdot \mathbf{E}_m + \hat{\mathbf{x}} \cdot \mathbf{Q} \tag{IV.20}$$

from here we will talk about the field amplitudes on the interface and the subscript 0 is dropped for conciseness. This boundary condition leads to

$$E_{dx} = E_{mx} + Q_x \quad (\text{IV.21})$$

or equivalently

$$E_d \cos \varphi_d = E_m \cos \varphi_m + Q_{\perp} \cos \varphi_q + Q_{\parallel} \cos \theta_q \quad (\text{IV.22})$$

where the subscripts d , m and q denote the angles are calculated using wavevectors K_d , K_m and q respectively.

Similarly, the boundary condition for B along y direction at $z = 0$ is

$$\hat{\mathbf{y}} \cdot \mathbf{B}_d = \hat{\mathbf{y}} \cdot \mathbf{B}_m \quad (\text{IV.23})$$

which give

$$E_d K_d = E_m K_m + q Q_{\perp} \quad (\text{IV.24})$$

Together with the two equations (IV.22) and (IV.24) from boundary conditions, the amplitude of electric fields on the interface can be solved

$$E_d = \frac{Q_{\perp}(K_m \cos \varphi_q - q \cos \varphi_m) + K_m Q_{\parallel} \cos \theta_q}{K_m \cos \varphi_a - K_d \cos \varphi_m} \quad (\text{IV.25a})$$

$$E_m = \frac{Q_{\perp}(K_d \cos \varphi_q - q \cos \varphi_d) + K_d Q_{\parallel} \cos \theta_q}{K_m \cos \varphi_a - K_d \cos \varphi_m} \quad (\text{IV.25b})$$

On the other hand, the Cartesian components of the electric fields on the interface can be achieved using the boundary conditions while considering the conversion

that

$$Q_x = Q_{\parallel} \cos \theta_q + Q_{\perp} \cos \varphi_q \quad (\text{IV.26a})$$

$$Q_z = Q_{\parallel} \sin \theta_q + Q_{\perp} \sin \varphi_q \quad (\text{IV.26b})$$

then the equation (IV.24) is now

$$K_{dz}E_{dx} - K_{dx}E_{dz} = K_{mz}E_{mx} - K_{mx}E_{mz} + q_z Q_x - q_x Q_z \quad (\text{IV.27})$$

which leads to

$$E_{dx}(\tan \theta_a - \tan \varphi_a) = E_{mx}(\tan \theta_m - \tan \varphi_m) + (Q_x \tan \theta_q - Q_z) \quad (\text{IV.28})$$

The equations of the boundary conditions (IV.21) and (IV.28) can be solved to get the x components of the fields:

$$E_{dx} = \frac{Q_x(\tan \theta_q - \tan \theta_m + \tan \varphi_m) - Q_z}{(\tan \theta_d - \tan \varphi_d) - (\tan \theta_m - \tan \varphi_m)} \quad (\text{IV.29a})$$

$$E_{mx} = \frac{Q_x(\tan \theta_q - \tan \theta_d + \tan \varphi_d) - Q_z}{(\tan \theta_d - \tan \varphi_d) - (\tan \theta_m - \tan \varphi_m)} \quad (\text{IV.29b})$$

And the z components of the electric field can be achieved via the relations that $E_z = E_x \tan \varphi$ accordingly. Up to this point, we achieved the solutions of the third harmonic generation problem to the differential equations (IV.8a) and (IV.8b).

Conversion Efficiency and Propagation Angles

The propagation direction, the angle of propagation ϑ relative to the x axis, of the free wave in the THG is determined by its wavevector K_m via

$$\tan \vartheta = \frac{\text{Re}[K_{mz}]}{\text{Re}[K_{mx}]} \quad (\text{IV.30})$$

When the imaginary part of K_m is small, the two angles θ and ϑ are approximately the same since $|K_m| \sim \text{Re}[K_m]$.

For quantitative illustration as well as for simplicity, we use the metal dielectric function for silver given by the Drude model as an example,

$$\varepsilon(\omega) = \varepsilon_\infty - \frac{\omega_p^2}{\omega^2 + i\gamma_p\omega}. \quad (\text{IV.31})$$

where for silver [54] the background dielectric constant $\varepsilon_\infty = 5.1$, the plasma frequency $\omega_p = 8.65\text{eV}$ and the damping constant $\gamma_p = 0.0045\text{eV}$. Then the expression of the propagation angle can be written explicitly in terms of the incident frequency ω of the SPP wave:

$$\begin{aligned} \tan \vartheta &= \frac{K_{mz}}{K_{mx}} = \frac{\sqrt{K_M^2 - q_x^2}}{q_x} = \frac{\sqrt{K_M^2 - (3k_x(\omega_1))^2}}{3k_x(\omega_1)} \\ &= \frac{\sqrt{\omega_3^2 \varepsilon_m(\omega_3) - 9\omega^2 \frac{\varepsilon_m(\omega_1)}{\varepsilon_d(\omega_1) + \varepsilon_m(\omega_1)}}}{3\omega \sqrt{\frac{\varepsilon_m(\omega_1)}{\varepsilon_d(\omega_1) + \varepsilon_m(\omega_1)}}} \\ &= \frac{\sqrt{\varepsilon_\infty - \frac{\omega_p^2}{9\omega_1^2 + 3i\omega_1\gamma_p} - \frac{\varepsilon_\infty - \frac{\omega_p^2}{\omega_1^2 + i\omega_1\gamma_p}}{\varepsilon_d(\omega_1) + \varepsilon_\infty - \frac{\omega_p^2}{\omega_1^2 + i\omega_1\gamma_p}}}}{\sqrt{\frac{\varepsilon_\infty - \frac{\omega_p^2}{\omega_1^2 + i\omega_1\gamma_p}}{\varepsilon_d(\omega_1) + \varepsilon_\infty - \frac{\omega_p^2}{\omega_1^2 + i\omega_1\gamma_p}}}} \end{aligned} \quad (\text{IV.32})$$

here ω_1 is the incident SPP wave frequency and $\omega_3 = 3\omega_1$ is the tripled frequency. When the incident SPP frequency ranges within the optical range, e.g. $400\text{nm} \sim 700\text{nm}$, the radiation angle of the THG ranges from 46° to 56° , pointing off from

the interface into the metal, as indicated in Fig. 4.3.

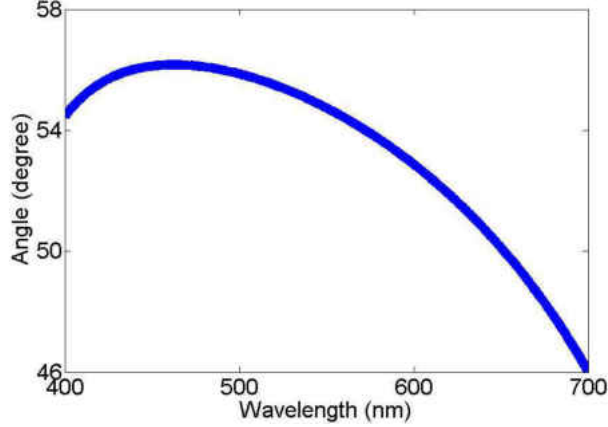


Figure 4.3 A plot of the propagation angle. The propagation angle of the THG v.s. the incident SPP wavelength. The angle is analytically determined by (IV.32).

With all the field strength relations available, we may now discuss the conversion efficiency from the incident SPP wave to the propagating THG signal wave. Here we choose the incident SPP to have a field amplitude of $E_x = 10^4 V/m$, which is proven to not violate the non-depletion approximation as we will see later. This amplitude also corresponds to a $1W$ continuous laser focused onto a $1mm^2$ spot. The incident wavelength of the SPP is chosen to be $690nm$ for convenience in later calculations.

First, we get the magnetic field for SPPs considering they are TM mode[74]

$$H_y = \frac{B_y}{\mu_0} = \varepsilon_0 c^2 B_y = \frac{\varepsilon_0 c^2}{\omega} (k_z E_x - k_x E_z), \quad (IV.33)$$

The energy density of the SPPs are given in [74]. In the dielectric, there is no

dispersion, so on the surface of the interface,

$$u_d = \frac{1}{4} [\varepsilon_0 E \cdot E^* + \mu_0 H \cdot H^*] \quad (\text{IV.34})$$

where

$$E \cdot E^* = E_0 e^{-i(\omega t - \mathbf{k} \cdot \mathbf{r})} \cdot E_0^* e^{+i(\omega t - \mathbf{k}^* \cdot \mathbf{r})} = (E_x E_x^* + E_z E_z^*) e^{-2[\text{Im}(k_x)x + \text{Im}(k_z)z]} \quad (\text{IV.35})$$

and

$$H \cdot H^* = H_y H_y^* e^{-2[\text{Im}(k_x)x + \text{Im}(k_z)z]} \quad (\text{IV.36})$$

The total energy (per unit area) associated with the surface polaritons is determined by integration over z , the energy per unit surface area being

$$U_d = \int_0^\infty u_d dz = \frac{u_d}{8\text{Im}k_{az}} = \frac{1}{8\text{Im}k_{az}} [\varepsilon_0 (E_x E_x^* + E_z E_z^*) + \mu_0 H_y H_y^*] \quad (\text{IV.37})$$

For SPPs in a metal, the effective energy density on the surface is

$$u_m = \frac{1}{4} \text{Re} \left[\frac{d(\omega\varepsilon)}{d\omega} \right]_{\omega} \mathbf{E}(\mathbf{x}) \cdot \mathbf{E}^*(\mathbf{x}) + \frac{1}{4} \mu_0 \mathbf{H}(\mathbf{x}) \cdot \mathbf{H}^*(\mathbf{x}) \quad (\text{IV.38})$$

And the total energy density in the metal (per unit area) is

$$U_m = \int_{-\infty}^0 u_m dz \quad (\text{IV.39})$$

The total energy over all space is

$$U_{total} = U_m + U_d \quad (\text{IV.40})$$

The third harmonic wave has the same expressions for the energy densities except for a tripled frequency ω_3 .

Assuming the metal/dielectric interface is between silver and vacuum such that $\varepsilon_d = 1$, then the conversion efficiency from SPP to THG wave is

$$\eta_{eff} = \frac{U_m(\omega_3)}{U_{total}(\omega_1)} \sim 0.1\% \quad (\text{IV.41})$$

This result is consistent with the non-depletion approximation that the energy drained from SPP wave by THG signal is small enough that the SPP wave remains relatively unaltered.

Adjustment to Realistic Noble Metals

According to the Drude model, silver is absorptive at wavelength $\lambda = 690nm$, but it would be transparent when frequency is tripled at wavelength $\lambda = 230nm$, which is known as the ultraviolet transparency for metals. However, the noble metals have increasingly strong absorption due to interband transitions, as seen in Fig. 2.1b.

For the case of silver, however, we found in literature data a very narrow transparency window just above the plasma frequency in the wavelength range of $293nm \sim 314nm$, so that within this wavelength, the real part of the dielectric function is positive (transparent) while the imaginary part (absorption) is not dominating so a portion of the generated THG wave can still penetrate and propagate [37]. In order to have the THG wave result in the transparency window, the incident SPP wavelength should be tuned to the infrared range

of $880\text{nm} \sim 941\text{nm}$. This corresponding frequency range of the SPPs and the resultant transparency window frequency range are shown in Fig. 4.4.

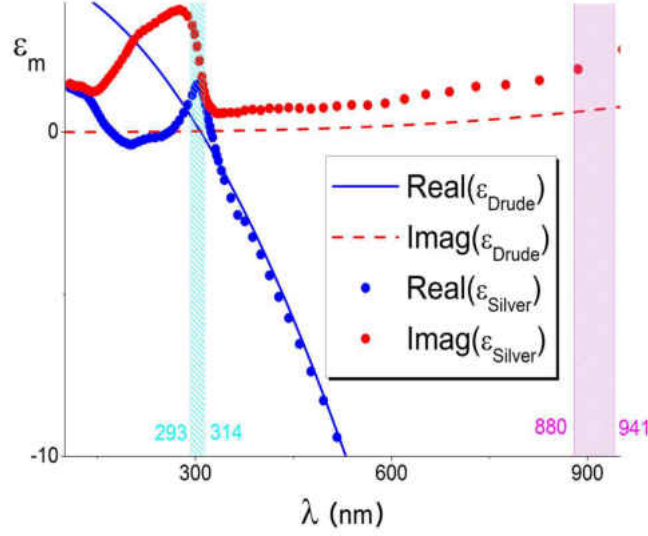


Figure 4.4 Drude model vs realistic metals. Close agreement is shown between Drude model and real data for the dielectric function of silver. The shaded green represents the transparency window and the (dense) shaded pink represent corresponding suggested wavelength of the incident SPPs. The literature data is cited from [37].

To reduce the absorption of the THG wave in silver, a metal film can be used. When a silver film of 50nm thickness is used, the generated THG wave has 35% of the amplitude (10% of the intensity) transmitted through the metal film.

The interband transition of silver also alters the calculation in the propagation angle of the THG in metal, via the realistic data of $\epsilon_m(\omega)$. The $\vartheta - \lambda_{spp}$ curve has an obvious red shift but keeps the basic shape similar to the analytical one in Drude

model. The resultant THG wave can be easily separated from the incident SPPs since the radiation direction of the THG wave clearly steers away at a rather large angle relative to the SPPs, as shown in the shaded area in Fig. 4.5.

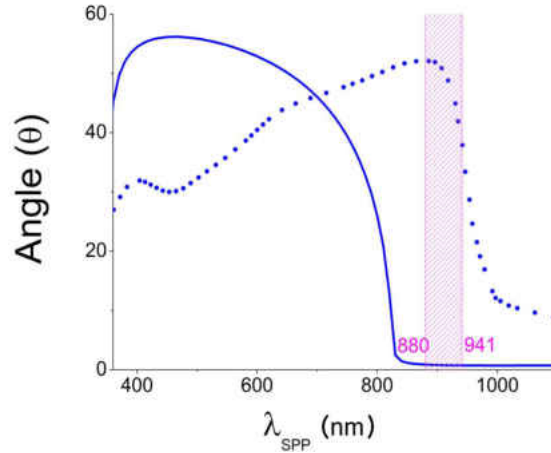


Figure 4.5 A plot of propagation angle with silver data. The angle of propagation for the THG in metal is altered from the case with the Drude model. The solid line is the angle predicted with the Drude model and the dotted curve is angles calculated from literature data of silver. The shaded area indicates the suggested wavelength for the incident SPP. Using the suggested wavelength, the resultant THG will fall into the transparency window of silver and the radiation angle is shown in dots in the graph.

Conclusions

This chapter demonstrated that a third harmonic generation (THG) wave is induced by SPPs in a nonlinear noble metal. The energy conversion efficiency is

calculated to be about 0.1% with the intensity of a common used laser in labs, e.g. with amplitude $E \sim 10^4 V/m$. When the frequency of the induced THG wave is above the plasma frequency of the metal, the THG wave is predicted to radiate through the metal if the Drude model is adapted for optical properties of the metal. For realistic metals, however, the radiation of the THG wave in the metal is strongly absorbed at and above the plasma frequency. A special case among the noble metals is silver. Silver, according to the experimental data in the literature, has a narrow ultraviolet transparency window around the plasma frequency. Our calculation for silver shows reasonable transmittance of the THG wave through a $50nm$ film and the direction of THG radiation is clearly separated from the incident SPPs by its propagation angle.

CHAPTER V

PLASMONIC SOLITONS

Introduction to Optical Solitons

This work was unpublished. Dr. Miriam Deutsch set up the proposed frame of the work I was the primary contributor to all the derivations and analysis of the results.

A solitary wave, often referred to as a soliton, describes a special kind of nonlinear wave, which keeps its wave properties unchanged, such as the pulse shape or beam diameter, during propagation in a medium. This kind of stable solitary wave can exist only with the presence of nonlinearities in the host medium. The spatial properties of a wave are changed because of the wave nature: a wave always diffracts during propagation. The temporal properties of a wave are changed, such as pulse broadening, if there is any dispersion present in the host dispersive medium. The spatial diffraction and temporal broadening, however, can be compensated in a medium with nonlinear response. Thus it is possible to achieve a stable wave form, a soliton, when the broadening and the diffraction are completely counteracted by nonlinear effects [75]. The fundamental mathematical models of solitons were built

in the 1960s [76] [77], and the concept of solitons has since evolved to a much broader range in different areas of physics [78][79][80][81][82][83].

In the context of nonlinear optics, one very important example of optical solitons, which is widely used in the field of modern communication technology, is a stable optical pulse guided in an optical fiber [84]. This stable optical pulse is formed by the balance of the (nonlinear) optical Kerr effect in the fiber and the material's (linear) chromatic dispersion [85][86]. The optical Kerr effect is a direct result from the third order susceptibility $\chi^{(3)}$ of the glass, and this optical Kerr effect in the fiber is represented by the change of the optical refractive index Δn , which is proportional to the light intensity, I .

Optical solitons are classified as being temporal or spatial according to the confinement types of the optical waves. For light with temporal confinement such as an optical pulse, the chromatic dispersion, e.g. high frequency wave components travel faster than low frequency components, introduces a frequency chirp and therefore leads to broadening in the pulse. Due to different phase velocities of frequencies in the optical pulse, frequency components are redistributed such that higher frequencies are in the leading half of the pulse and lower frequencies in the trailing half. In a nonlinear medium, this chirp may be compensated by the optical Kerr effect, because the nonlinear interaction between the electric field and the medium causes a down-shift of the frequencies in the leading half of the pulse, and an up-shift in the trailing half. These two opposing effects, in some special cases,

can be balanced and thus the pulse keeps its amplitude envelope unchanged in time and this is called an optical *temporal soliton*, as shown in Fig. 5.1.

For an optical beam in space, an analogy to the pulse can be used, where the confinement now is in one or more transverse spatial dimensions instead of in time for an optical pulse. A light beam is usually diffracted, spreading the beam diameter during the wave propagation. The nonlinear refractive index here acts as a focusing lens and induces a self-focusing (or self-defocusing) effect with a positive (or negative) Kerr coefficient [83]. When the self-focusing counteracts and cancels the effect of diffraction, the light beam may maintain its diameter during propagation in space, and it is called an optical *spatial soliton* as shown in Fig. 5.2 [40].

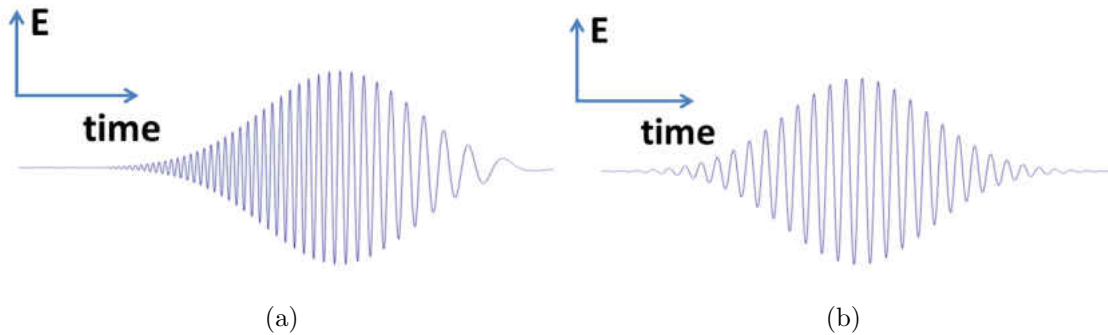


Figure 5.1 A schematic of a temporal soliton formed by the balance between linear dispersion and optical Kerr effect. (a) The optical pulse is chirped and broadened by dispersion in a linear medium. (b) The dispersion can be balanced to form an unchanged wave packet by the optical Kerr effect.

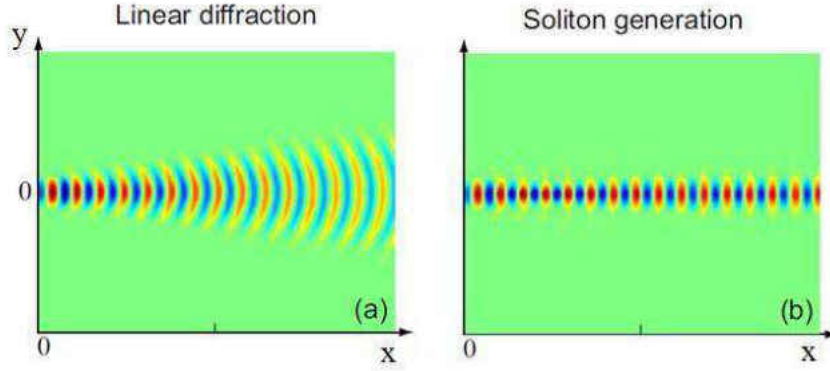


Figure 5.2 A schematic of a spatial soliton formed by the balance between the diffraction and self-focusing. (a) The optical beam diffracts in a linear medium while propagating along x -axis. (b) The diffraction is counteracted by the self-focusing effect since the refractive index is higher towards the center of the optical beam. (Figure taken from ref. [83])

The first experimental observation of optical temporal solitons were discovered in optical fibers in 1980 by Mollenauer *et al.* using pico-second laser pulses [84]. When an optical pulse propagates in an optical fiber, the chromatic dispersion induces a group-velocity dispersion (GVD) and the nonlinear refractive index leads to a self-phase modulation (SPM). The GVD is the group velocity dependency on angular frequencies, namely the derivative of the inverse group velocity, v_g , with respect to the angular frequency, ω :

$$\text{GVD} \equiv \frac{\partial}{\partial \omega} \frac{1}{v_g} = \frac{\partial^2 \beta}{\partial \omega^2} \quad (\text{V.1})$$

where β is the propagation wavevector. The SPM is a nonlinear phase delay, $\Delta\phi(t)$, which arises from the varying optical refractive index $\Delta n(I)$, where I is the optical

intensity. In an optical pulse, this phase delay leads to a change of the instantaneous frequency, $\omega(t)$, according to $\Delta\omega(t) = \frac{d\Delta\phi(t)}{dt}$. Consequently the change of the instantaneous frequency is relative to the pulse instantaneous intensity by [87]:

$$\Delta\omega(t) \sim -\frac{\partial\Delta n(I(\mathbf{r}, t))}{\partial t} \quad (\text{V.2})$$

For a medium with optical Kerr effect such that $\Delta n \propto I$, (V.2) gives $\Delta\omega(t) < 0$ for the leading half of the pulse where the instantaneous intensity $I(t)$ is ramping up, and $\Delta\omega(t) > 0$ for the trailing half. At the communication wavelength, $1.55\mu m$, the silica-glass fiber has an anomalous GVD ($\text{GVD} < 0$) and a positive Kerr effect ($\chi^{(3)} > 0$). An anomalous GVD means the group velocity is faster for higher frequencies and slower for lower frequencies in an optical pulse. The anomalous GVD gives the optical pulse a down-chirp, namely the instantaneous frequency of the pulse decreases with time, as indicated in Fig. 5.1(a). At the same time, according to (V.2), the SPM will tune down frequency when the intensity increases, which is the case for the leading half of the optical pulse, and tune up frequency when the intensity decreases, which is the case for the the trailing half. With a carefully chosen pulse envelope (a *sech* function), the two competing effects over the frequencies can completely cancel each other and a temporal soliton is achieved, as shown in Fig. 5.1(b).

A spatial soliton is formed in a scenario similar to a temporal soliton. On one hand, the wave nature of an optical beam tends to diffract the beam diameter during propagation in a medium, as indicated in Fig. 5.2(a). On the other hand,

with the presence of the positive Kerr effect ($\chi^{(3)} > 0$), the center of the beam, which of high optical intensity, induces larger optical refractive index than the rim area of the beam. This index inhomogeneity works in a similar way to a convex lens that focuses a light beam, thus results a self-focusing effect of the light beam, as shown in Fig. 5.2(b). For a beam with a properly chosen initial shape and intensity, the self-focusing of the beam may exactly counteract the diffraction so that the optical pulse remains a confined, self-trapped beam. Thus a spatial soliton is achieved [88] [89].

The traditional wave-packet (temporal) solitons discussed in previous sections are called bright solitons since they are positive energy packages against background of zero field intensity, as shown in Fig. 5.3a. When the GVD is normal ($\text{GVD} > 0$), however, a bright soliton is not supported by the nonlinear optical fibers discussed above. Instead, a special “dark” pulse, which exists as an absence of energy on a continuous wave (CW) background, turns out to be stable and thus called a *dark* soliton, as shown in Fig. 5.3b. With a positive Kerr effect ($\chi^{(3)} > 0$), the bright temporal solitons can be achieved in a nonlinear medium with an anomalous GVD ($\text{GVD} < 0$) while that the dark temporal solitons can be achieved with a normal GVD ($\text{GVD} > 0$). For the case of spatial optical beams, diffraction plays a role similar to the GVD in the temporal soliton case: the self-focusing nonlinearity from positive Kerr effect ($\chi^{(3)} > 0$) gives bright spatial solitons and the self-defocusing nonlinearity from negative Kerr effect ($\chi^{(3)} < 0$) gives dark spatial solitons.

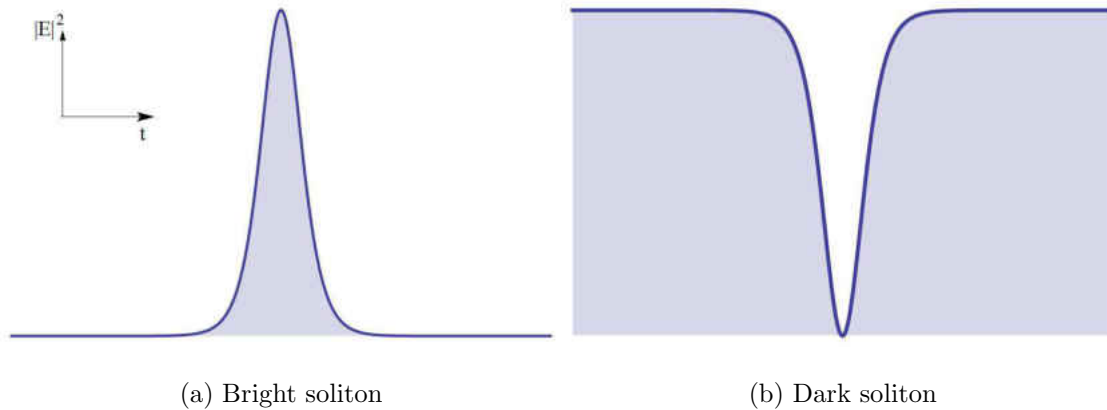


Figure 5.3 A bright soliton and a dark soliton. (a) A bright temporal soliton is an optical wave packet which maintains the envelope against zero background. It can be achieved when the nonlinear medium has an anomalous GVD ($GVD < 0$). The envelope function for the intensity envelope function of the bright soliton is a $|sech|^2$ function. (b) A dark temporal soliton is a stable “dark” hole which keeps its shape against non-zero background. It can be achieved when the nonlinear medium has a normal GVD. The intensity of a dark soliton is a $|tanh|^2$ function.

The general theory on solitons has been developed long before the observation of the optical solitons and applied to a variety of different nonlinear systems [76] [90] [77]. There are a number of partial differential equations (PDEs) with soliton solutions in different forms and a few may be solved exactly, such as the Kortveg-de Vries (KdV) equation, the sine-Gordon equation and the nonlinear Schrödinger (NLS) equation [75][91]. For a light-wave packet traveling in a waveguide, the evolution of the envelope is commonly described by a NLS equation [86], which

results in an optical temporal soliton solution. The general solutions of the NLS equation can be achieved by the inverse scattering transform [92], a method for nonlinear systems similar to the Fourier transform in linear systems and the solutions are known to be solitons.

In this dissertation, we discuss nonlinear SPPs guided by a planar structure, which is an interface between a semi-infinite metal with Kerr nonlinearities and a semi-infinite linear dielectric. We focus on the temporal equation for the envelope of an SPP pulse, and find the conditions for a soliton-type solution. Early works on spatial SPP solitons are discussed in detail by pioneer researchers in [83], and the solitons on the surface comprised of a nonlinear dielectric and a linear metal are discussed in [93]. The early works about SPP solitons focused on the dielectric nonlinearity but by and large ignores the metal nonlinearity. However, as we stated in previous chapters, the metal nonlinearity is comparable to or even greater than the nonlinearity in many dielectrics such as glass. With the negative permittivity of a metal, nonlinearities in the metal may lead to different phenomena than nonlinearities in a dielectric. In this chapter, our research is presented to reveal the soliton properties while the metal nonlinearity is taken into account.

The Temporal Nonlinear Schrödinger Equation of SPPs

The linear SPP is a two-dimensional guided wave propagating at the interface

between a dielectric and a metal. In this section we study the new features of the nonlinear SPPs when the metal nonlinearity is taken into account.

We consider only the third order nonlinear susceptibility $\chi^{(3)}$ in the metal, but ignore its sum frequency effect, such as was discussed in Ch.IV in the sections concerning THG, because the sum frequency effect occurs at very low efficiency. Therefore the metal is regarded as a Kerr medium and the nonlinear polarization \mathbf{P}^{NL} in (III.5) has the same frequency as the original incident light \mathbf{E}

$$\mathbf{P}^{NL}(\omega) = \varepsilon_0 \chi^{(3)}(\omega = \omega + \omega - \omega) \mathbf{E}(\omega) \mathbf{E}(\omega) \mathbf{E}(\omega) \quad (\text{V.3})$$

Combining the time derivatives in the nonlinear wave equation(III.4) gives

$$\nabla \times \nabla \times \mathbf{E} + \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \varepsilon_M \mathbf{E} = 0 \quad (\text{V.4})$$

where $\varepsilon_M = \varepsilon_m + \varepsilon_{NL}$ is the total dielectric function of the metal, including both the linear contribution $\varepsilon_m = 1 + \chi^{(1)}$, and the nonlinear contribution $\varepsilon_{NL} = \chi^{(3)} |E|^2$, at response frequency ω . Here we prefer using the optical dielectric function ε of the metal instead of the optical refractive index n even when n is commonly used in the analysis for nonlinear fiber optics [86] [94]. We do this because ε can be negative or complex in our discussion and the traditional definition of $n = \sqrt{\varepsilon}$ often leads to inconvenience and confusion.

The divergence free equation $\nabla \cdot \mathbf{E} = 0$ holds inside the isotropic metal except at the interface, such that $\nabla \times \nabla \times \mathbf{E} = -\nabla^2 \mathbf{E}$. Consequently the Cartesian components are not coupled in (V.4) with the chosen coordinates shown in Fig. 2.2

and (V.4) can be written in component form:

$$\hat{x} : -(\partial_x^2 + \partial_y^2 + \partial_z^2)E_x + \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \varepsilon_M E_x = 0 \quad (\text{V.5a})$$

$$\hat{z} : -(\partial_x^2 + \partial_y^2 + \partial_z^2)E_z + \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \varepsilon_M E_z = 0 \quad (\text{V.5b})$$

where $\varepsilon_0 \mu_0 = 1/c^2$. Considering the TM nature of linear SPP profile, E_x and E_z have a fixed relation $E_x = \zeta E_z$ where ζ is a piecewise constant in the dielectric and in the metal. Since the two equations are linearly dependent, we may consider E_z first under the scalar approximation to eliminate the complexity of the vector nature of the nonlinear wave equation. The equation derivation process for E_x is identical to E_z . Apply Fourier transform

$$E_z(\mathbf{r}, t) = \frac{1}{\sqrt{2\pi}} \int \tilde{E}_z(\mathbf{r}, \omega - \omega_0) e^{-i(\omega - \omega_0)t} dt \quad (\text{V.6})$$

upon equation (V.5b), with ω_0 being the central frequency of the incident SPP wave, and it gives

$$\nabla^2 \tilde{E}_z + \frac{\omega^2}{c^2} \varepsilon_M \tilde{E}_z = 0 \quad (\text{V.7})$$

The tilde “ \sim ” represents that the quantity is a function of frequency ω instead of time t . Separating the propagation factor $e^{i\beta_0 x}$ from \tilde{E}_z (where β_0 being the propagation constant to be determined from the eigensolution of the equation) the electric field \tilde{E}_z can be written as

$$\tilde{E}_z(x, y, z, \omega - \omega_0) = F_m(y, z) \tilde{A}_z(x, \omega - \omega_0) e^{i\beta_0 x} \quad (\text{V.8})$$

Here $F_m(y, z)$ and $\tilde{A}_z(x, \omega - \omega_0)$ are the transverse and longitudinal functions

respectively. The differential equation (V.7) is now

$$\tilde{A}_z(\partial_y^2 + \partial_z^2)F_m + F_m(\partial_x^2\tilde{A}_z + 2i\beta_0\partial_x\tilde{A}_z - \beta_0^2\tilde{A}_z) + \frac{\omega^2}{c^2}\varepsilon_M F_m\tilde{A}_z = 0 \quad (\text{V.9})$$

For a long, smooth optical pulse the slowly varying amplitude approximation $|\partial_x^2\tilde{A}_z| \ll |\beta_0\partial_x\tilde{A}_z|$ is satisfied so that the higher order term $\partial_x^2\tilde{A}_z \simeq 0$ is ignored. Applying the standard method of variable separation to the partial differential equation (V.9) yields

$$\frac{(\partial_y^2 + \partial_z^2)F_m}{F_m} + \varepsilon_M \frac{\omega^2}{c^2} = -\frac{2i\beta_0\partial_x\tilde{A}_z - \beta_0^2\tilde{A}_z}{\tilde{A}_z} \equiv \tilde{\beta}^2 \quad (\text{V.10})$$

where $\tilde{\beta}$ is an undetermined coefficient and it can be a function of ω .

The differential equation for E_z in the dielectric medium has a similar expression:

$$\frac{(\partial_y^2 + \partial_z^2)F_d}{F_d} + \varepsilon_d \frac{\omega^2}{c^2} = -\frac{2i\beta_0\partial_x\tilde{A}_z - \beta_0^2\tilde{A}_z}{\tilde{A}_z} \equiv \tilde{\beta}^2 \quad (\text{V.11})$$

with \tilde{A}_z, β_0 and $\tilde{\beta}$ the same as required by the boundary conditions, but with a different transverse functions F_d and a different dielectric function ε_d . We can make the further assumptions that the SPP wave extends to $\pm\infty$ along the y axis, and that the derivatives with respect to y are negligible. Therefore we have a reduced 2-dimensional problem with x and z dependencies only. Consequently F_d and F_m are functions of z only and the partial derivatives against z become total derivatives. The two equations (V.10) and (V.11) on different sides of the interface

can be combined by writing F and ε_T as piecewise functions

$$F(z) \equiv \begin{cases} F_d(z) & z \geq 0 \quad (\text{Dielectric}) \\ F_m(z) & z < 0 \quad (\text{Metal}) \end{cases} \quad (\text{V.12})$$

$$\varepsilon_T(z, \omega) \equiv \begin{cases} \varepsilon_d(\omega) & z \geq 0 \quad (\text{Dielectric}) \\ \varepsilon_M(\omega) & z < 0 \quad (\text{Metal}) \end{cases} \quad (\text{V.13})$$

Here ε_M is the nonlinear dielectric function of the metal and possesses the property that $\text{Re}\varepsilon_M < 0$ and $|\text{Re}\varepsilon_M| \gg |\text{Im}\varepsilon_M|$. The parameter ε_d is the dielectric constant, which is close to unity and $|\varepsilon_M| \gg |\varepsilon_d| \gtrsim 1$. For partial differential equations, the standard procedure to achieve the common coefficient $\tilde{\beta}$ is to solve the eigenfunctions:

$$\frac{d^2 F}{dz^2} - \left[\tilde{\beta}^2 - \varepsilon_T \frac{\omega^2}{c^2} \right] F = 0 \quad (\text{V.14a})$$

$$2i\beta_0 \frac{\partial \tilde{A}_z}{\partial x} + (\tilde{\beta}^2 - \beta_0^2) \tilde{A}_z = 0 \quad (\text{V.14b})$$

We obtain the linear solution of function F by letting $\varepsilon_M = \varepsilon_m$ and use it as the first order approximation for the nonlinear case to achieve the function \tilde{A}_z and thus the first equation in (V.14) is written explicitly as a piecewise eigenequation

$$\frac{d^2 F}{dz^2} = \begin{cases} \left(\tilde{\beta}^2 - \varepsilon_d \frac{\omega^2}{c^2} \right) F & z \geq 0 \\ \left(\tilde{\beta}^2 - \varepsilon_m \frac{\omega^2}{c^2} \right) F & z < 0 \end{cases} \quad (\text{V.15})$$

where ε_m is the linear dielectric function for the metal.

Now consider the physical restrictions that the waves from both sides match at the boundary and decay at ∞ . When $\varepsilon_m < 0$, this is possible only if $\tilde{\beta}^2 > \varepsilon_d \frac{\omega^2}{c^2}$.

Thus the boundary condition for E_z gives the transverse solution:

$$F(z) = \begin{cases} \varepsilon_m e^{-z\sqrt{\tilde{\beta}^2 - \varepsilon_d \frac{\omega^2}{c^2}}} & z \geq 0 \\ \varepsilon_d e^{+z\sqrt{\tilde{\beta}^2 - \varepsilon_m \frac{\omega^2}{c^2}}} & z < 0 \end{cases} \quad (\text{V.16})$$

which is exactly the linear solution for SPP stated in previous chapters.

The boundary condition for E_x , on the other hand, leads to

$$\left. \frac{dF}{dz} \right|_{z=0^+} = \left. \frac{dF}{dz} \right|_{z=0^-} \quad (\text{V.17})$$

resulting again in the linear SPP dispersion relation

$$\tilde{\beta}^2 = \frac{\omega^2}{c^2} \frac{\varepsilon_m \varepsilon_d}{\varepsilon_m + \varepsilon_d} \equiv \tilde{\beta}_L^2 \quad (\text{V.18})$$

For the the nonlinear case, an ansatz is achieved by perturbing $\tilde{\beta}$, to the first order, from the linear $\tilde{\beta}_L$ such that now $\tilde{\beta} \equiv \tilde{\beta}_L + \Delta\tilde{\beta}$. The two photon absorption effect of bulk noble metals is not fully studied but known to be very weak[95], and it is not directly related to our topic. To keep the physics simple and succinct, we ignore the two photon absorptions related to the $\chi^{(3)}$ and keep $\chi^{(3)}$ a pure real number. The discussion, however, can be extended to a $\chi^{(3)}$ with both real and imaginary parts with nontrivial conversions [96].

It is useful to Taylor expand $\tilde{\beta}_L$ about the carrier frequency ω_0

$$\tilde{\beta}_L(\omega) = \beta_0 + (\omega - \omega_0)\beta_1 + \frac{1}{2}(\omega - \omega_0)^2\beta_2 + \frac{1}{6}(\omega - \omega_0)^3\beta_3 + \dots \quad (\text{V.19})$$

and keep terms up to the second order, where the constant $\beta_0 = \tilde{\beta}_L|_{\omega=\omega_0}$ is the propagation constant at carrier frequency ω_0 and the other coefficient constants are defined as

$$\beta_n \equiv \left. \frac{d^n \tilde{\beta}_L}{d\omega^n} \right|_{\omega=\omega_0} \quad (\text{V.20})$$

where $\beta_1 = 1/v_g$ is the inverse of the group velocity, and β_2 is known as the group velocity dispersion, or GVD.

With approximation to the first order, the right hand side of (V.14b) is

$$\tilde{\beta}^2 - \beta_0^2 \simeq 2\beta_0(\tilde{\beta} - \beta_0) \quad (\text{V.21})$$

and the equation is written as

$$\frac{\partial \tilde{A}_z}{\partial x} = i \left[\tilde{\beta}_L(\omega) + \Delta\tilde{\beta} - \beta_0 \right] \tilde{A}_z \quad (\text{V.22})$$

Now using the Taylor expansion of $\tilde{\beta}_L$ and keeping second order accuracy,

$$\frac{\partial \tilde{A}_z}{\partial x} = i \left[(\omega - \omega_0)\beta_1 + \frac{1}{2}(\omega - \omega_0)^2\beta_2 + \Delta\tilde{\beta} \right] \tilde{A}_z \quad (\text{V.23})$$

Using an inverse Fourier transform

$$A_z(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \tilde{A}_z(x, \omega - \omega_0) e^{-i(\omega - \omega_0)t} d\omega \quad (\text{V.24})$$

on the equation above leads to the nonlinear schrödinger equation (NLS) equation for the longitudinal slowly varying amplitude A

$$\frac{\partial A_z}{\partial x} = -\beta_1 \frac{\partial A_z}{\partial t} - \frac{i}{2}\beta_2 \frac{\partial^2 A_z}{\partial t^2} + i\Delta\beta A_z \quad (\text{V.25})$$

A similar equation can be derived for the A_x component, too. Using the linear relation that $E_x = \zeta E_z$, the combination of the equations for A_z and A_x gives the NLS equation for the amplitude A

$$\frac{\partial A}{\partial x} = -\beta_1 \frac{\partial A}{\partial t} - \frac{i}{2} \beta_2 \frac{\partial^2 A}{\partial t^2} + i \Delta \beta A \quad (\text{V.26})$$

where $A = |\mathbf{A}|$ and $\mathbf{A} = \hat{z}A_z + \hat{x}A_x$.

This is the equation that governs the evolution of the amplitude of the SPP, taking into account the Kerr effect of the supporting metal. It is the starting point of plasmonic solitons and gives temporal soliton solutions. The coefficients β_1 and β_2 are approximated to the first order by derivatives of the linear $\tilde{\beta}_L$ with respect to angular frequency. $\Delta\beta$ is the perturbation to the linear wave factor with the presence of $\chi^{(3)}$ nonlinearity. It is under further consideration in next section. In addition the attributes of the soliton solutions are discussed in detail in the following sections.

Nonlinear SPP Dispersion Relations

In this section, we are going to determine the general properties of the coefficients β_1 , β_2 and $\Delta\beta$ in the NLS equation (V.26) from the last section. For a TM wave guided along the x direction with propagation factor $e^{-i(\omega t - \tilde{\beta}x)}$, the nonlinear dispersion relation is derived from the field relation with the nonlinear

dielectric function, ε_M , in the nonlinear metal [83] [97] [98]:

$$\frac{dE_x}{dz} = -\frac{i}{\tilde{\beta}} \left(\frac{\omega^2}{c^2} \varepsilon_M - \tilde{\beta}^2 \right) E_{mz} \quad (\text{V.27a})$$

$$\frac{d}{dz} (\varepsilon_M E_{mz}) = -i \tilde{\beta} \varepsilon_M E_x \quad (\text{V.27b})$$

$$H_y = -\frac{\omega \varepsilon_0 \varepsilon_M}{\tilde{\beta}} E_{mz} \quad (\text{V.27c})$$

We can get a first integral from the above relation equations:

$$\left(\frac{dE_x}{dz} \right)^2 = \frac{\omega^2}{c^2} \left[\left(\varepsilon_M - \frac{c^2 \tilde{\beta}}{\omega^2} \right) E_{mz}^2 - \varepsilon_M E_x^2 - \chi^{(3)} (E_x E_{mz})^2 + \frac{\chi^{(3)}}{2} (E_x^4 + E_{mz}^4) \right] \quad (\text{V.28})$$

In the next step we aim to obtain the nonlinear SPP dispersion relation. Using (V.27a) for the left-hand side in equation (V.28) above:

$$\left(1 - \frac{\varepsilon_M c^2}{\tilde{\beta}^2 \omega^2} \right) E_{mz}^2 = \left(\varepsilon_M - \frac{c^2 \tilde{\beta}}{\omega^2} \right) E_{mz}^2 - \varepsilon_M E_x^2 - \chi^{(3)} (E_x E_{mz})^2 + \frac{\chi^{(3)}}{2} (E_x^4 + E_{mz}^4) \quad (\text{V.29})$$

To eliminate ε_M , the transverse profile and boundary conditions of the E field are applied. In the dielectric, the transverse dependency is $E(z) \sim e^{-\kappa_d z}$ where the decay factor $\kappa_d = \sqrt{\tilde{\beta}^2 - \varepsilon_d \frac{\omega^2}{c^2}}$, while the subscript d labels components in the dielectric. Then the divergence free equation for the displacement field $\nabla \cdot \mathbf{D} = 0$ gives $E_{dz} = i \tilde{\beta} E_x / \kappa_d$.

Considering components of the electric displacement, $D_{dz} = D_{mz}$, leads to $\varepsilon_d E_{dz} = \varepsilon_M E_{mz}$. On the interface it gives

$$\varepsilon_M = i \varepsilon_d \frac{\tilde{\beta}}{\kappa_d} \frac{E_x}{E_{mz}} \quad (\text{V.30})$$

Using the expression for ε_M , (V.29) is now

$$\frac{\chi^{(3)}}{2}E_x^4 - \left(\varepsilon_M + \frac{c\varepsilon_d}{\omega\kappa_d}\right)E_x^2 - i\frac{\varepsilon_d\tilde{\beta}}{\kappa}E_xE_{mz} - \frac{\chi^{(3)}}{2}E_{mz}^4 = 0 \quad (\text{V.31})$$

where all the E fields here take the values at the surface ($z = 0$). This equation determines the nonlinear SPP dispersion relation since $\tilde{\beta}$ is implicitly determined by ω , E_x and E_{mz} . It is easy and straightforward to verify that it complies with the linear dispersion relation by letting $\chi^{(3)} = 0$ and $\varepsilon_M = \varepsilon_m$.

For a general nonlinear dispersive metal, an additional relation may be required between E_x and E_{mz} . We first consider an ideal case that the losses in the metal are small and negligible. This gives a $\pi/2$ phase shift between E_{mz} and E_x , thus we can define a real-valued total amplitude $E_T > 0$ with

$$E_T^2 = |E_T|^2 \equiv |E_{mz}|^2 + |E_x|^2 = E_{mz}^2 - E_x^2 \quad (\text{V.32})$$

since the total field has the form $\hat{\mathbf{z}}E_{mz} + i\hat{\mathbf{x}}E_x$. Together with

$$\varepsilon_M = i\frac{\varepsilon_d\tilde{\beta}}{\kappa_d}\frac{E_x}{E_{mz}} \quad (\text{V.33})$$

the first two relations are achieved by squaring (V.33), and gives the third relation

is given by multiplying (V.33) with E_{mz}^2 :

$$E_x^2 = -E_T^2\frac{(\varepsilon_M)^2\kappa_d^2}{\varepsilon_d^2\tilde{\beta}^2 + (\varepsilon_M)^2\kappa_d^2} \quad (\text{V.34a})$$

$$E_{mz}^2 = E_T^2\frac{\varepsilon_d^2\tilde{\beta}^2}{\varepsilon_d^2\tilde{\beta}^2 + (\varepsilon_M)^2\kappa_d^2} \quad (\text{V.34b})$$

$$E_xE_{mz} = -iE_T^2\frac{\varepsilon_d\varepsilon_M\tilde{\beta}\kappa_d}{\varepsilon_d^2\tilde{\beta}^2 + (\varepsilon_M)^2\kappa_d^2} \quad (\text{V.34c})$$

Finally we express all field components in terms of E_T^2 . We use $\kappa_d^2 = \tilde{\beta}^2 - \varepsilon_d \frac{\omega^2}{c^2}$ in (V.31) to solve for the nonlinear dispersion relation of $\tilde{\beta}$:

$$\tilde{\beta}^2 = \frac{\omega^2}{c^2} \frac{\varepsilon_M \varepsilon_d \left(\varepsilon_M - \varepsilon_d - \frac{E_T^2 \chi^{(3)}}{2} \right)}{[(\varepsilon_M)^2 - \varepsilon_d^2] - \frac{E_T^2 \chi^{(3)}}{2} [(\varepsilon_M)^2 + \varepsilon_d^2]} \quad (\text{V.35})$$

The nonlinear dielectric function of the metal $\varepsilon_M = \varepsilon_m + \chi^{(3)} E_T^2$ can be inserted into (V.35) for a more explicit expression. Care must be taken when one tries to achieve $\tilde{\beta}$ and operates a square root on the right hand side, because the square root operation requires a proper cut in the complex plain so that the condition $\text{Re} \tilde{\beta} > 0$ holds [99]. Using a Taylor expansion identity with respect to the small quantity $\delta = \chi^{(3)} E_T^2$:

$$(\varepsilon_m + \delta) \sqrt{\frac{a - \delta}{gx + b}} = \sqrt{\frac{a}{b}} \left[\varepsilon_m + \left(1 - \frac{\varepsilon_m}{2a} - \frac{g\varepsilon_m}{2b} \right) \delta \right] + O(\delta^2) \quad (\text{V.36})$$

where the intermediate notations $a = 2(\varepsilon_d - \varepsilon_m)$, $b = 2\varepsilon_m(\varepsilon_m^2 + \varepsilon_d^2)$ and $g = 3\varepsilon_d^2 - 5\varepsilon_m^2$ are used. To the first order of $\chi^{(3)} E_T^2$ we have

$$\tilde{\beta} = \frac{\omega}{c} \sqrt{\frac{\varepsilon_d \varepsilon_m}{\varepsilon_m + \varepsilon_d}} - \frac{\omega}{c} \chi^{(3)} E_T^2 \frac{\varepsilon_d}{4(\varepsilon_m + \varepsilon_d)} \sqrt{\frac{\varepsilon_d}{\varepsilon_m(\varepsilon_m + \varepsilon_d)}} \quad (\text{V.37})$$

The expression of ε_m is often known from tabulated data, or an analytic model, e.g. Drude model as discussed in Ch.II, and we can calculate

$$\Delta\beta \equiv (\tilde{\beta} - \tilde{\beta}_L)|_{\omega=\omega_0} = \frac{\omega_0}{c} \chi^{(3)} E_T^2 \sqrt{\frac{\varepsilon_m^3}{16\varepsilon_m(\varepsilon_m + \varepsilon_d)^3}} \Big|_{\omega=\omega_0} \quad (\text{V.38})$$

For simplicity, we define the nonlinear factor

$$\nu = \frac{\omega_0}{c} \chi^{(3)} \sqrt{\frac{\varepsilon_m^3}{16\varepsilon_m(\varepsilon_m + \varepsilon_d)^3}} \Big|_{\omega=\omega_0} \quad (\text{V.39})$$

Considering that $E_T \sim Ae^{-i\omega t} e^{i\beta x}$, the NLS equation (V.26) is now written as

$$\frac{\partial A}{\partial x} = -\beta_1 \frac{\partial A}{\partial t} - \frac{i}{2} \beta_2 \frac{\partial^2 A}{\partial t^2} + i\nu |A|^2 A \quad (\text{V.40})$$

The other coefficients β_1 and β_2 are given by evaluating to the first order using the linear dispersion relation at $\omega = \omega_0$

$$\beta_1 \equiv \left. \frac{d\tilde{\beta}_L}{d\omega} \right|_{\omega=\omega_0} \quad \text{and} \quad \beta_2 \equiv \left. \frac{d^2\tilde{\beta}_L}{d\omega^2} \right|_{\omega=\omega_0} \quad (\text{V.41})$$

Analytical and Numerical Solutions of Dark Solitons

Dark Soliton Solutions

In a lossless metal, the expression for ε_m from the Drude model is

$$\varepsilon_m = \frac{\omega_p^2}{\omega^2} - \varepsilon_\infty \quad (\text{V.42})$$

where ε_∞ is the background dielectric constant at low frequency. All coefficients in the NLS equation are real quantities and explicitly they are written as functions of frequency ω :

$$\tilde{\beta}_L = \frac{\omega}{c} \sqrt{\varepsilon_d} \sqrt{\frac{\omega_p^2/\omega^2 - \varepsilon_\infty}{\omega_p^2/\omega^2 - \varepsilon_\infty - \varepsilon_d}} \quad (\text{V.43a})$$

$$\Delta\beta_L = \frac{\omega_0}{c} \chi^{(3)} E_T^2 \sqrt{\frac{\varepsilon_d^3}{16(\omega_p^2/\omega^2 - \varepsilon_\infty)(\omega_p^2/\omega^2 - \varepsilon_\infty - \varepsilon_d)^3}} \quad (\text{V.43b})$$

It is clear that $\Delta\beta_L > 0$ if $\chi^{(3)} > 0$, which is consistent with the fact that the larger (nonlinear) refractive index leads to a bigger wavevector. The β_1 and β_2 are calculated and graphed as in Fig. 5.4. Note that below the plasma frequency

ω_p , the quantities β_1 , β_2 and $\Delta\beta$ are all positive real numbers with lossless Drude model (V.42).

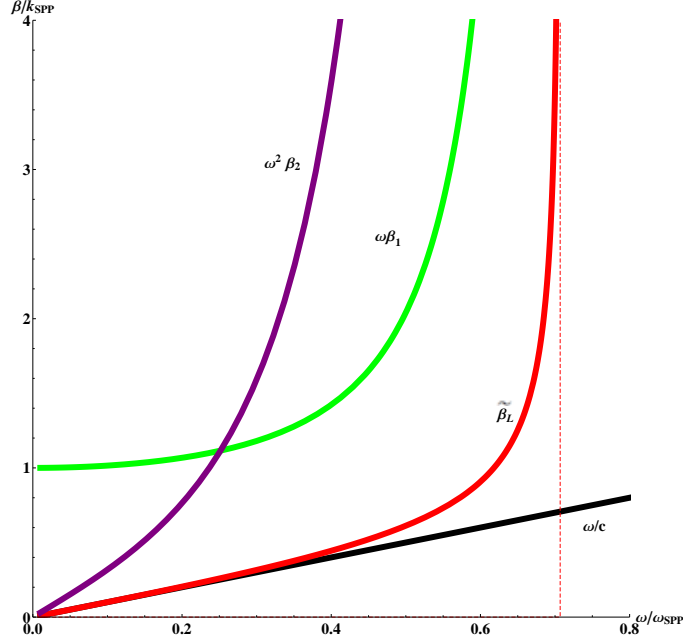


Figure 5.4 The coefficients in the NLS equation. The coefficients β_1 and β_2 in the NLS equation are calculated from the linear SPP dispersion relation using lossless Drude model. β_1 and β_2 are positive below plasma frequency ω_p . The vertical dashed line is the asymptote of $\tilde{\beta}_L$, which represents the frequency limit of a linear SPP wave.

To solve the NLS equation (V.40) analytically, we transform $A(x, t) =$

$A(\xi(x), \tau(x, t))$ by defining

$$\tau = (t - \beta_1 x)/T_0 \quad (\text{V.44a})$$

$$\xi = x/L_D \quad (\text{V.44b})$$

$$u = A\sqrt{|\nu|L_D} \quad (\text{V.44c})$$

where T_0 is the duration of the incident SPP pulse, $L_D = T_0^2/|\beta_2|$ is the SPP dispersion length due to GVD and $|\nu|$ is the absolute value of ν . Then we have a NLS equation with standard form

$$i\frac{\partial u}{\partial \xi} - \frac{1}{2} \frac{\beta_2}{|\beta_2|} \frac{\partial^2 u}{\partial \tau^2} + \frac{\nu}{|\nu|} |u|^2 u = 0 \quad (\text{V.45})$$

For SPPs on noble metal surface, $\chi^{(3)} \equiv \text{Re}[\chi^{(3)}] > 0$ so that $\nu > 0$. Also, with normal group-velocity dispersion $\text{GVD} \equiv \frac{\partial^2 \beta}{\partial \omega^2} > 0$ so that $\beta_2 > 0$. We have now

$$i\frac{\partial u}{\partial \xi} - \frac{1}{2} \frac{\partial^2 u}{\partial \tau^2} + |u|^2 u = 0 \quad (\text{V.46})$$

and this standard form of a NLS equation has a dark temporal soliton solution [86].

It is conventional to use a slightly different standard form with a positive second order derivative term [92] [86]. By defining $X = -\xi$ and $U(X, \tau) = u(\xi, \tau)$, the NLS equation turns to

$$i\frac{\partial U}{\partial X} + \frac{1}{2} \frac{\partial^2 U}{\partial \tau^2} - |U|^2 U = 0 \quad (\text{V.47})$$

The most general form of the solution is

$$U(X, \tau) = U_0 B_2 \tanh[U_0 B_2 (\tau - B_1 U_0 X)] + i B_1 e^{-iV_0^2 X + i\theta_0} \quad (\text{V.48})$$

where B_1, B_2 and θ_0 are arbitrary constants which depend on initial conditions. For a dark soliton, the boundary condition is

$$|U|_{\xi \rightarrow \pm\infty} = U_0 \quad (\text{V.49})$$

where U_0 is the background wave amplitude.

The coefficients B_1 and B_2 have a constraint that $B_1^2 + B_2 = 1$. Thus we may write $B_1 = \sin \phi$ and $B_2 = \cos \phi$ where the variable ϕ represents the “darkness” of the soliton. For example when $\sin \phi = 0$ the minimum intensity of the solution goes to 0 and we call it a “dark” soliton. When $\sin \phi \neq 0$, the minimum intensity of the soliton is greater than 0 and the solution forms a “gray” soliton.

Switching back to the lab frame of reference with parameters t and x , the solution is

$$A(x, t) = A_0 \left\{ \cos \phi \tanh \left[\frac{A_0 \sqrt{\nu} \cos \phi}{\sqrt{\beta_2}} t - \left(\frac{\sqrt{\nu} \beta_1 - A_0 \nu \sin \phi}{\sqrt{\beta_2}} \right) A_0 \cos \phi x \right] + i \sin \phi \right\} e^{i A_0^2 \nu x + i \theta_0} \quad (\text{V.50})$$

And the normalized intensity is

$$\left| \frac{A(x, t)}{A_0} \right|^2 = 1 - \cos^2 \phi \operatorname{sech}^2 \left[\frac{A_0 \sqrt{\nu} \cos \phi}{\sqrt{\beta_2}} t - A_0 \cos \phi \left(\frac{\sqrt{\nu} \beta_1}{\sqrt{\beta_2}} + A_0 \nu \sin \phi \right) x \right] \quad (\text{V.51})$$

To satisfy this eigenfunction, the pulse duration, pulse length and group velocity

are required to be

$$\tau_0 = \frac{\sqrt{\beta_2}}{A_0\sqrt{\nu}\cos\phi} \quad (\text{V.52a})$$

$$L_0 = \frac{\sqrt{\beta_2}}{(\sqrt{\nu}\beta_1 - A_0\nu\sin\phi\sqrt{\beta_2})A_0\cos\phi} \quad (\text{V.52b})$$

$$v_g = \frac{1}{\beta_1 - A_0\sqrt{\nu}\sin\phi\sqrt{\beta_2}} \quad (\text{V.52c})$$

To illustrate the core physics, the simplest case is chosen, i.e. $\phi = 0$ and $\theta_0 = 0$ in (V.50) so the dark soliton has an amplitude and intensity equations

$$A(x, t) = A_0 \tanh \left[\frac{A_0\sqrt{\nu}}{\sqrt{\beta_2}}(t - \beta_1 x) \right] e^{iA_0\nu x} \quad (\text{V.53a})$$

$$|A|^2 = |A_0|^2 \left\{ 1 - \operatorname{sech}^2 \left[\frac{A_0\sqrt{\nu}}{\sqrt{\beta_2}}(t - \beta_1 x) \right] \right\} \quad (\text{V.53b})$$

Here A_0 is the peak amplitude of the optical soliton. For the soliton, the peak amplitude is not a free parameter but determined by the duration τ_0 and other constants as in (V.52a). In fact, (V.52a) determines the fundamental amplitude of a soliton for a given duration τ_0 by

$$A_0 = \frac{\sqrt{\beta_2}}{\tau_0\sqrt{\nu}\cos\phi}. \quad (\text{V.54})$$

Given a pulse duration, if the initial peak amplitude is slightly different to the fundamental amplitude in (V.54), then the pulse will adjust itself during propagation and eventually reach the stable soliton amplitude given by (V.54). If the initial peak amplitude is very low, the nonlinear effect can not counteract the linear dispersion and the pulse will be simply linearly dispersed. If the initial

peak amplitude is very high, complicated behaviors such as high-order soliton and soliton break-up could occur [92] [91] [75].

Numerical Analysis with and without Metal Loss

The existence of the dark SPP soliton for the nonlinear SPP is demonstrated by the numerical solution to the NLS equation (V.46) where the amplitude of the electric field is already normalized to the stable soliton solution amplitude A_0 , and in the initial condition we also set the arbitrary phase $\phi = 0$. As shown in Fig. 5.5, where the loss in the metal is ignored, the SPP pulse propagates along the x -direction on the metal surface while its temporal intensity profile is conserved over distance.

When the amplitude-duration relation in (V.54) is not satisfied, the optical pulse is not an eigensolution to the NLS equation. We show numerically it will either break down to separate pulses for an initial pulse with larger amplitude, as shown in Fig. 5.6a, or oscillate during propagation for a pulse with smaller amplitude, as shown in Fig. 5.6b. In the case of larger amplitude, two small “gray” pulses appear symmetrically on each side of the main “dark” pulse, with the main pulse width narrowed down. In the case of small amplitude, small continuous oscillations propagate out of the center pulse on both sides. The initial shape of the fundamental “dark” pulse is a \tanh function, which is an odd function, or odd symmetry. Therefore, the main pulse keeps “dark”, i.e., center intensity being

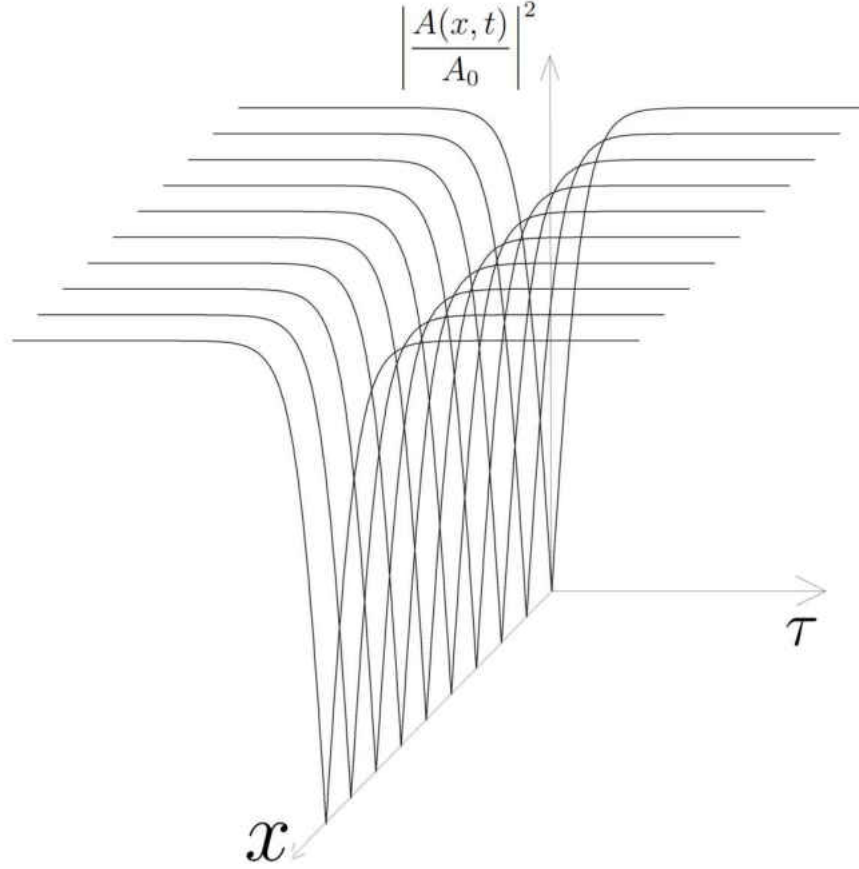


Figure 5.5 The stability of the soliton presented by numerical calculations. The temporal profile of the SPP pulse does not change while propagating along the x -direction on the nonlinear metal surface. Thus a stable dark soliton is achieved.

zero, regardless of the initial conditions. For reference numerical values, the SPP related material constants used in the numerical calculation are listed in Tab.D.1 (for vacuum/silver) [41].

In our numerical solution for the lossless metal model, an SPP pulse with $10ps$ duration is chosen at the communication wavelength $1.55\mu m$. The peak amplitude is calculated by the threshold equation (V.52a), which may be attained using a

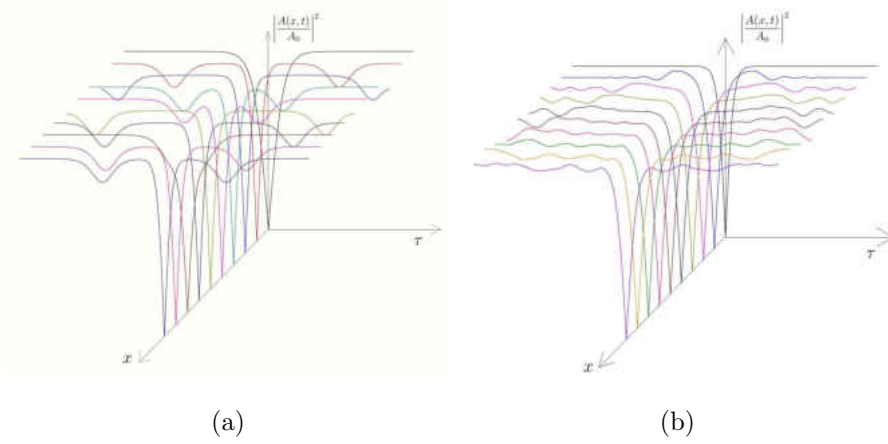


Figure 5.6 A schematic of pulse propagation when the soliton condition is not satisfied. (a) The optical pulse breaks up when the initial amplitude is greater than the threshold. [100] (b) The optical pulse has an oscillation when the initial amplitude is less than the threshold.

pulsed laser at an average power of $0.5W$ with 80 MHz repetition value. The pulse evolution for different initial amplitudes is shown in Fig. 5.5 and Fig. 5.6, and the numerical parameters are shown in Tab.D.1.

We now turn to the more realistic case of a noble metal with Ohmic loss. The noble metal has finite absorption in optical range, thus a linear lossy term is added into the lossless NLS equation:

$$i \frac{\partial U}{\partial X} + \frac{1}{2} \frac{\partial^2 U}{\partial \tau^2} - |U|^2 U = i\Gamma U \quad (\text{V.55})$$

where $\Gamma \equiv L_D/(2\beta_1)$ is a linear loss parameter representing the loss effect in the

metal [86]. Here L_D is the SPP dispersion length and β_1 is the inverse of the SPP wavevector.

For weak absorption where $\Gamma \ll 1$, it is common to use perturbation method to analytically approximate the result [101]. For the case of an SPP pulse on a silver surface, however, the loss term can not be always regarded weak and the numerical solution is the only way to demonstrate the evolution of the SPP pulse. The numerical recipe of Crank Nicolson (CN) is used and for details one can refer to Appendix D for the numerical algorithm and Appendix E for the source code written in MatLab[®].

To account for the metal loss more precisely, we use here a modified Drude model for silver:

$$\varepsilon_m(\omega) = \varepsilon_\infty - \frac{\omega_p^2}{\omega^2 + i\gamma_p\omega} + i\frac{\sigma}{\varepsilon_0\omega} \quad (\text{V.56})$$

where γ_p is the damping coefficient and σ is the AC conductivity constant at optical frequency. The physical meanings of all symbols and their values for silver are summarized in table Tab.D.2 in Appendix D.

The SPP pulse duration is chosen to be $8fs$ so that the SPP linear dispersion effect, represented by L_D , and the nonlinear effect, represented by L_{NL} , are comparable to each other, both about $700\mu m$. The repetition rate is tuned down to $1Hz$ to avoid inadvertent heating caused by high peak intensities, in possible experimental setups. The distance in the simulation is $x = 2L_D$ and when the initial pulse is set to be exactly the soliton threshold for the lossless case, the pulse

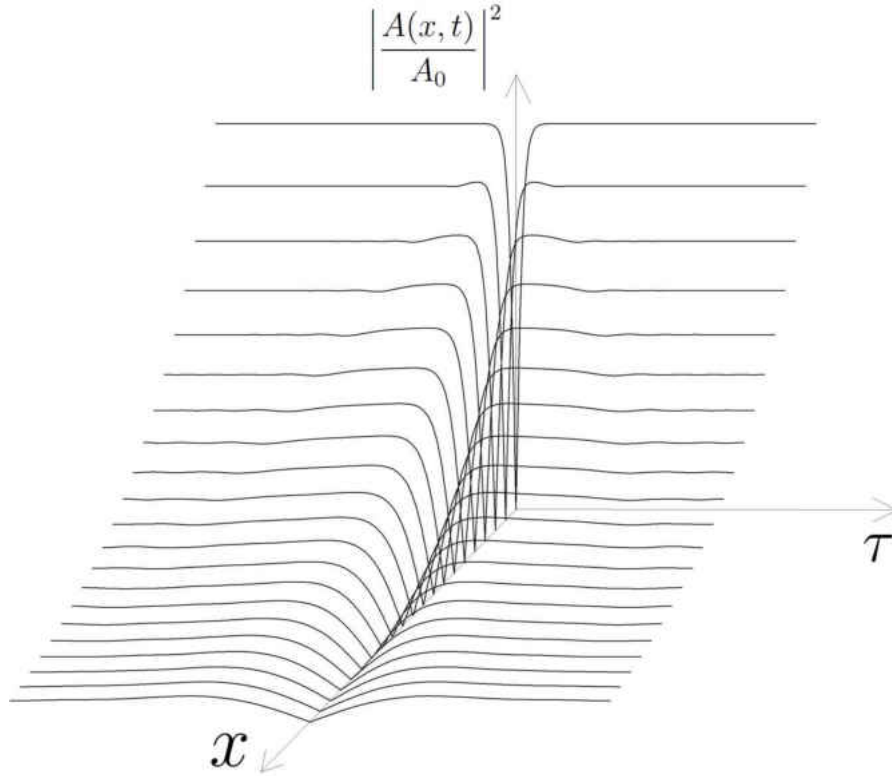


Figure 5.7 The evolution of a dark SPP pulse with loss. The dark SPP pulse travels on a metal surface with linear loss. Starting with a soliton profile, which is a stable solution in lossless case, the pulse decreases smoothly with no internal oscillations or break-ups.

is damped while propagating, but keeps a single pulse as shown in Fig. 5.7. With other input amplitude distribution, a pulse break-up or internal oscillation occurs, as shown in Fig. 5.8, due to the multiple breakup of high-order solitons and soliton modulation instability [86] [102]. When the initial pulse amplitude is greater than the soliton threshold, obvious pulse break-up occurs, similar to the lossless case in Fig. 5.6a. After a short propagation distance, however, the amplitude of the pulse

decreases due to the loss, and the pulse break-up gradually turns into oscillations as shown in Fig. 5.8a. When the initial pulse amplitude is less than the soliton threshold, small oscillations propagate on both sides of the main pulse as shown in Fig. 5.8b, similar to the lossless case in Fig. 5.6b.

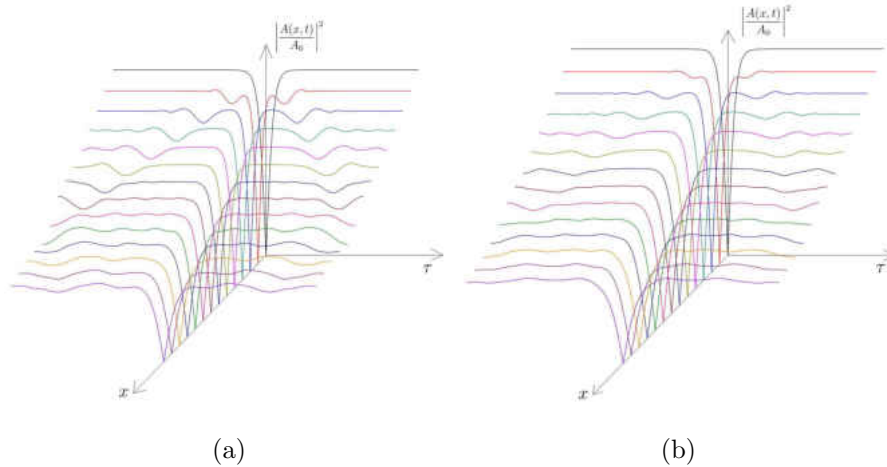


Figure 5.8 A schematic of pulse propagation with loss when the initial soliton condition is not satisfied. (a) The optical pulse breaks up when the initial amplitude is greater than the threshold. The pulse break-up is diminished during propagation and gradually turns into oscillations because of the amplitude decrease due to the loss (b) The optical pulse oscillates when the initial amplitude is less than the threshold.

The damping of the dark SPP pulse may be compensated to keep a stable soliton, by adding optical gains in the dielectric layer. Another possible way to

avoid the loss is by coherently coupling the dark SPP pulse to other plasmon modes, with an effect similar to Electromagnetically induced transparency (EIT) [103].

Conclusions

In this chapter, we achieved a nonlinear Schrödinger (NLS) equation for a temporal SPP pulse. The nonlinear dispersion relation of the temporal SPP pulse was derived. This SPP pulse was supported by a nonlinear metal and a linear dielectric. In the analytical studies, the nonlinear metal was dispersive and lossless, while the linear dielectric was chosen to be vacuum. An analytical solution of the NLS equation was achieved and was a temporal dark soliton for the nonlinear SPP pulse. The soliton was infinitely stable during propagation for the ideal case. To form this stable dark soliton, the relations among pulse peak amplitude, pulse amplitude outline, pulse duration and background amplitude were achieved. For a realistic metal with finite loss, numerical calculations were performed. With the initial pulse satisfying the lossless soliton condition, the pulse maintained a single recognizable peak when the amplitude was damped during propagation. With the initial pulse not satisfying the lossless soliton condition, the pulse suffered internal oscillation or pulse break-up for intensities too low or too high.

CHAPTER VI

CONCLUSIONS

The study of plasmonics has been drawing increased attention in recent years. The physics of surface plasmon polaritons offers new and efficient ways to guide light signals in nano-scale dimensions and at terahertz operation speeds using metallic nanostructures. It also brings new technologies, such as chemical sensing and biological sensing, to the other fields of science. To fully explore the potential advantages of SPPs with respect to fast operation speed, broad bandwidth and nanometer scale size, fundamental research into the nonlinear properties of SPPs has become an active field, with its own challenges and excitements. This dissertation serves to reveal a small tip of the iceberg.

We started this dissertation with a review of the history and recent developments of plasmonics. In Chapter II, we presented a complete and detailed derivation of major properties of linear surface plasmon polaritons (SPPs), such as the dispersion relations. This derivation is the starting point of the following chapters and it lays the foundation for our research. The SPPs were shown as a guided wave between a dielectric and a metal surface. The optical properties of the metals were analytically described by the Drude model when the frequencies of the light waves were below the plasma frequency. A simple layout of planar SPPs

was used and the vacuum was used for the dielectric layer. The SPP dispersion relations and the key dimensions of the SPPs were achieved.

In Chapter III, we introduced nonlinear optics in general then applied the nonlinear wave equations to our specific nonlinear SPP system. The noble metals in the SPP system possess third order nonlinearity, denoted with the third order susceptibility $\chi^{(3)}$. This $\chi^{(3)}$ lead to the nonlinear polarization term \mathbf{P}^{NL} and then a nonlinear wave equation for the metals was derived. In order to solve the nonlinear wave equation, two specific cases were considered: sum frequency generation and difference frequency generation. In the sum frequency generation case, the non-depletion approximation was used to decouple the tripled frequency components from the original frequency components. In the difference frequency generation case, a nonlinear diffractive index was introduced to take into account the fact that the generated frequency was indistinguishable from the original source frequency. For both cases, a designated orientation of the metal crystal lattice was chosen to reduce the nonlinear response of $\chi^{(3)}$ to a scalar.

In Chapter IV, the steady-state solution of the nonlinear wave equation with sum frequency generation was achieved. The solution showed third harmonic generation in the nonlinear SPP process. The induced tripled frequency signal from the SPPs is still bounded in the dielectric side of the interface. When the optical properties of the metal are described by the classical Drude model, this tripled frequency signal is able to propagate inside the metal with an obvious angle

of about 40° relative to the interface. The energy conversion efficiency from the SPP to the tripled signal was analytically calculated. The efficiency is about 0.1% using a standard continuous wave laser with visible light intensity $I = 10^3 W/cm^2$. This low conversion efficiency complies with the non-depletion approximation made when solving the nonlinear wave equations. The propagation angle of the tripled signal was also calculated and found to have a dependency on the permittivity of the metal and the incident wavelength of the SPP, and the signal is well separated in space by its propagation direction from the original SPP waves. The angle ranges from about 46° to about 56° when the incident wavelength is in the optical range. For realistic metals, such as noble metals with interband transitions, absorption above the plasma frequencies had to be considered. Tabulated data in literature shows a transparency window around the plasma frequency for silver, with reduced absorption, and consequently we used this transparency data to show possible improvements for experimental setups, e.g. we considered a silver film with finite thickness instead of a semi-infinite bulk. Within this transparency window, the tripled signal has a transmittance of about 10% for a $50nm$ silver film, and the propagation angle is about 40° .

In Chapter V, we investigated the temporal properties of nonlinear SPPs by considering the nonlinear Kerr effect in silver. First we derived the nonlinear dispersion relation from the boundary conditions involving metal nonlinearities. With the nonlinear dispersion relation, we applied slowly the varying amplitude

approximation and perturbed the wavevectors in the wave equations, and a temporal nonlinear Schrödinger (NLS) equation for the nonlinear SPP propagation was achieved. Given the values of the coefficients in the NLS equation, we found that the stable solution is a SPP pulse, and that the shape of the pulse possesses the form of a dark temporal soliton. The ideal dark soliton solution, when the metal is lossless, was also demonstrated by a numerical calculation. When losses in the metal accounted for, the numerical calculation showed that the evolution of the pulse depends strongly on its shape. When the initial shape of the pulse satisfies the same conditions of a soliton in the lossless case, the pulse propagated with only broadening and damping. Otherwise, the pulse suffered background oscillations or multi-peak breakups within a short distance after being launched.

The topics discussed in this dissertation are nonlinear plasmonics with the consideration of the metallic nonlinearities. Nonlinear SPPs have many more interesting properties to be explored. For future studies, we propose several improvements and directions of efforts. For the THG of nonlinear SPPs, the energy conversion efficiency can be increased by an incident SPP with higher intensity. The power draining effect of the source SPP has to be considered when the efficiency goes up where the non-depletion approximation is no longer valid, and the coupled wave equations with both fundamental and tripled frequencies have to be solved numerically. Without the non-depletion approximation, the propagation angle of the tripled frequency signal can have dependency on the input optical intensity. In

addition, both the conversion efficiency and the propagation angle of the tripled signal can be tunable if the dielectric layer is optically nonlinear or possesses optical gain. Different structures are worth attention too. The proposed dielectric-metal-dielectric (DMD) and metal-dielectric-metal (MDM) [104][105][106] layered structures, in which each layer may be linear or nonlinear, may add exciting new nonlinear effects in addition to what we have discussed in this dissertation. For instance, the dispersion relation of the tripled signal in the dielectric side within a single metal-dielectric interface may be altered by a nonlinear DMD or MDM structure due to the coherence and interference of the optical waves in different layers.

As for the SPP solitons, the main problem for achieving stable and long distance SPP solitons is material loss. Loss compensation has been a major direction of study in recent years, mostly in linear plasmonics [107]. A gain dielectric medium may be capable of compensating for the metal loss. When the dielectric layer has gain, the nonlinear boundary conditions are changed and thus leads to a revised dispersion relation of the SPP. It is possible, with a gain medium compensating the metal loss, that the nonlinear SPP solitons are stable and can propagate long distance. It may even be possible that the dark soliton solution is converted to a bright soliton solution, such as when the overall GVD is changed from normal to anomalous, by the gain dielectric medium.

Nonlinear plasmonics provide us powerful tools to manipulate and guide

optical signals. For instance, devices with nanometer scale and terahertz operation speed, as well as sub-wavelength digital signal transmission, may be utilized by nonlinear SPP processes discussed in this dissertation. For the full potential of the nonlinear plasmonics, it is necessary to carry out further systematic studies of the nonlinear plasmonics, in both theory and experiment. The future studies of the nonlinear plasmonics include, but not limited to, multi-layer structures such as MDM and DMD, patterned guiding surfaces and supporting dielectric materials with different optical properties such as optical nonlinearity and gain. Research on nonlinear plasmonics not only improves understanding of the fundamental nature of SPPs, it will also contribute novel methods to the fast growing technology world.

APPENDIX A

SOLUTION OF INHOMOGENEOUS SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS

In the section, we solve an inhomogeneous second order ordinary differential equation [108].

A second-order (inhomogeneous) ordinary linear differential equation of function $q(\tau)$ encountered often has the form

$$\frac{d^2q}{d\tau^2} + 2\zeta \frac{dq}{d\tau} + \omega_0^2 q = B e^{i\omega_s \tau} \quad (\text{A.1})$$

where B is an arbitrary constant and ω_s is the driving frequency.

The characteristic equation reads

$$r^2 + 2\zeta r + \omega_0^2 = 0 \quad (\text{A.2})$$

The full solution of (A.1) comprises two parts: $q = q_t + q_s$ where q_t is the general solution of the homogeneous equation, a.k.a the transient solution, while q_s is a particular solution, a.k.a. the steady-state solution.

For small damping $|\zeta| \ll 1$, the discriminant $\Delta = (2\zeta)^2 - 4\omega_0^2 < 0$, so the characteristic equation has two solutions that are complex conjugate of one another:

$$r_1 = -\zeta + i\sqrt{\omega_0^2 - \zeta^2} \quad (\text{A.3a})$$

$$r_2 = -\zeta - i\sqrt{\omega_0^2 - \zeta^2} \quad (\text{A.3b})$$

Then the general (transient) solution to the homogeneous differential equation is

$$q_t(\tau) = e^{-\zeta\tau} \left(C_1 e^{-i\tau\sqrt{\omega_0^2 - \zeta^2}} + C_2 e^{i\tau\sqrt{\omega_0^2 - \zeta^2}} \right) \quad (\text{A.4})$$

where C_1 and C_2 are determined by initial conditions. Note that $q_t \rightarrow 0$ when $t \rightarrow \infty$ no matter how small the damping ζ is.

The particular (steady-state) solution q_s can be found by the method of undetermined coefficients. By making ansatz $q_s(\tau) = \mathcal{A} e^{i\omega_s\tau}$ where \mathcal{A} is a complex amplitude to be determined. By substituting q_s back to (A.1) one finds $\mathcal{A} = B/(-\omega_s^2 + 2i\zeta\omega_s + \omega_0^2)$. The particular solution thus is oscillatory

$$q_s(\tau) = \frac{B e^{i\omega_s\tau}}{-\omega_s^2 + 2i\zeta\omega_s + \omega_0^2} \quad (\text{A.5})$$

In a real experiment, the damping coefficient is always finite, which gives

$$q(\tau) \xrightarrow{\tau \rightarrow \infty} q_s(\tau) \quad (\text{A.6})$$

Therefore, in the long time limit, only the steady-state solution is detected since the general solution dies out soon.

APPENDIX B

HELMHOLTZ THEOREM AND HELMHOLTZ EQUATION

The Helmholtz theorem states [109]:

Any sufficiently smooth, rapidly decaying vector field can be resolved into irrotational (curl-free) and solenoidal (divergence-free) component vector fields.

Given such a vector field $\mathbf{V}(\mathbf{r})$ satisfying

$$[\nabla \cdot \mathbf{V}]_{\infty} = 0 \tag{B.1a}$$

$$[\nabla \times \mathbf{V}]_{\infty} = 0 \tag{B.1b}$$

then there exist a scalar potential ϕ and a vector potential \mathbf{A} such that

$$\begin{aligned} \mathbf{V}(\mathbf{r}) &= -\nabla\phi + \nabla \times \mathbf{A} \\ &\equiv \mathbf{V}_{\parallel}(\mathbf{r}) + \mathbf{V}_{\perp}(\mathbf{r}) \end{aligned} \tag{B.2}$$

where \mathbf{V}_{\parallel} is the longitudinal (irrotational, curl-free) part that

$$\nabla \times \mathbf{V}_{\parallel} = 0 \tag{B.3}$$

and \mathbf{V}_{\perp} is the transverse (solenoidal, divergence-free) part that

$$\nabla \cdot \mathbf{V}_{\perp} = 0 \tag{B.4}$$

Applying Fourier transformation on vector field \mathbf{V} from \mathbf{r} to \mathbf{k} :

$$\tilde{\mathbf{V}}(\mathbf{k}) = \mathcal{F}[\mathbf{V}(\mathbf{r})] = \frac{1}{(\sqrt{2\pi})^3} \iiint_{\mathbf{r}} \mathbf{V}(\mathbf{r}) e^{i\mathbf{k}\cdot\mathbf{r}} d^3\mathbf{r} \tag{B.5}$$

gives

$$\mathbf{k} \times \tilde{\mathbf{V}}_{\parallel}(\mathbf{k}) = 0 \quad (\text{B.6a})$$

$$\mathbf{k} \cdot \tilde{\mathbf{V}}_{\perp}(\mathbf{k}) = 0 \quad (\text{B.6b})$$

and consequently

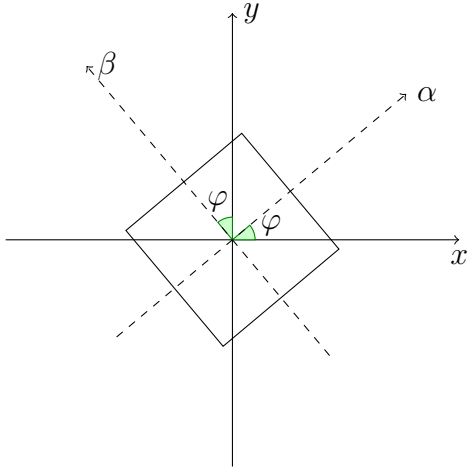
$$\begin{aligned} \nabla \times \nabla \times \mathbf{V}(\mathbf{r}) &\implies \\ i\mathbf{k} \times [i\mathbf{k} \times \tilde{\mathbf{V}}(\mathbf{k})] & \\ = i\mathbf{k} \times [i\mathbf{k} \times \tilde{\mathbf{V}}_{\perp}(\mathbf{k})] & \\ = i\mathbf{k} [i\mathbf{k} \cdot \tilde{\mathbf{V}}_{\perp}(\mathbf{k})] - (i\mathbf{k})^2 \tilde{\mathbf{V}}_{\perp}(\mathbf{k}) & \\ = - (i\mathbf{k})^2 \tilde{\mathbf{V}}_{\perp}(\mathbf{k}) & \quad (\text{B.7}) \end{aligned}$$

APPENDIX C

THE ROTATION MATRIX

The nonlinear polarization is induced by the incident fields via the nonlinear susceptibility $\chi^{(3)}$ tensor as in (III.18). The first three terms on the right hand of (III.18) are invariant under rotation thus independent of spatial coordinates and the sum is labeled as \mathbf{P}^I where the superscript indicates “isotropic”. The last term on the right hand of (III.18) is where the tensor nature of the $\chi^{(3)}$ imbedded. It is labeled as \mathbf{P}^A indicating “anisotropy” which is coordinate specific and defined as

$$P_i^A \equiv (M :: \mathbf{E}^a \mathbf{E}^b \mathbf{E}^c)_i = \sum_{jkl} M_{jkl}^i E_j^a E_k^b E_l^c \quad (\text{C.1})$$



$$\begin{aligned} E_\alpha &= E_x \cos \varphi + E_y \sin \varphi \\ E_\beta &= -E_x \sin \varphi + E_y \cos \varphi \\ \text{and} \\ P_x^A &= P_\alpha^A \cos \varphi - P_\beta^A \sin \varphi \\ P_y^A &= P_\alpha^A \sin \varphi + P_\beta^A \cos \varphi \end{aligned}$$

Figure C.1 The rotation matrix for (simplified 2-D) *crystal - lab* coordinates conversion. The conversion relations for \mathbf{E} and \mathbf{P} are explicitly represented.

where $ijkl$ represent any set of orthogonal coordinate axes. Now the nonlinear polarization is separated to isotropic and anisotropic parts:

$$\mathbf{P}^{NL} = \mathbf{P}^I + \mathbf{P}^A \quad (\text{C.2})$$

Practically it is convenient to write both \mathbf{P} and \mathbf{E} in lab reference frame, so the explicit form of tensor M in the lab coordinate frame is desired, and our central goal of this section is to find the expressions for the tensor M .

Firstly we deal with the rotation part in the tensor M and add the effect of $\chi^{(3)}$ later. It is easier to break the total rotation matrix K between the *cubic* and the *lab* frames to two steps: the rotation matrix T between the *cubic* and the *crystal* frames, and the rotation matrix R between the *crystal* and the *lab* frames:

$$\mathbf{E}_{x'y'z'} = R\mathbf{E}_{xyz} \quad (\text{C.3a})$$

$$\mathbf{E}_{\alpha\beta\gamma} = T^{-1}\mathbf{E}_{x'y'z'} \quad (\text{C.3b})$$

$$\mathbf{P}_{x'y'z'} = T\mathbf{P}_{\alpha\beta\gamma} \quad (\text{C.3c})$$

$$\mathbf{P}_{xyz} = R^{-1}\mathbf{P}_{x'y'z'} \quad (\text{C.3d})$$

Here \mathbf{E} is the electric field vector, \mathbf{E}_{xyz} is the vector \mathbf{E} expressed in lab frame, $\mathbf{E}_{x'y'z'}$ is the vector \mathbf{E} expressed in the crystal frame, and $\mathbf{E}_{\alpha\beta\gamma}$ is the vector \mathbf{E} expressed in the *cubic* frame. The the rotation matrix R between *lab* and *crystal* frames with rotation angle φ is defined as

$$R = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{C.4})$$

while the representation of rotation matrix T between the *cubic* and the *crystal* frames depends on the orientation of cubic symmetry axes:

$$T_{(111)} = \begin{pmatrix} \sqrt{\frac{2}{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix} \quad (\text{C.5a})$$

$$T_{(110)} = \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{pmatrix} \quad (\text{C.5b})$$

$$T_{(001)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{C.5c})$$

The subscripts here indicate the orientation of the crystal, e.g. the direction of the *crystal* z' axis, while the actual crystal “cut” or “shearing plane” is in a plane normal to this chosen axis, indicated as the shaded planes in Fig. C.4 Fig. C.3 and Fig. C.2. The T is derived by simple geometric relations, for example the (111) orientation gives:

$$\begin{aligned} \hat{\mathbf{x}}' &= \sqrt{\frac{2}{3}}\hat{\mathbf{x}} - \frac{1}{\sqrt{6}}(\hat{\mathbf{y}} + \hat{\mathbf{z}}) \\ \hat{\mathbf{y}}' &= \frac{1}{\sqrt{2}}(\hat{\mathbf{y}} - \hat{\mathbf{z}}) \\ \hat{\mathbf{z}}' &= \frac{1}{\sqrt{3}}(\hat{\mathbf{x}} + \hat{\mathbf{y}} + \hat{\mathbf{z}}) \end{aligned} \quad (\text{C.6})$$

where we have chosen this definition so that the new x axis is projected on to the original crystal axis in the plane of the crystal surface, shown in Fig. C.2.

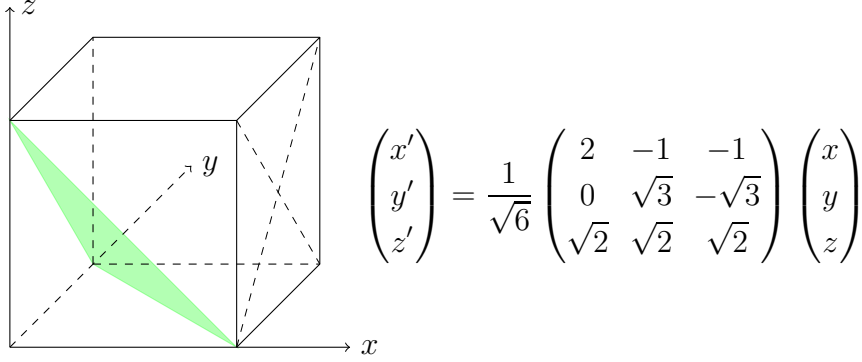


Figure C.2 The relation between (111) crystal “cut” (shaded) and the lab coordinates. The direction of the crystal orientation, \mathbf{z}' , is defined as the normal to the shaded surface, while the other axes \mathbf{x}' and \mathbf{y}' are chosen with a degree of freedom as long as the three axes comprise a right handed coordinate system. The cubic coordinates are not shown here.

For the (110) surface we have

$$\begin{aligned}\hat{\mathbf{x}}' &= \frac{1}{\sqrt{2}}(\hat{\mathbf{y}} - \hat{\mathbf{x}}) \\ \hat{\mathbf{y}}' &= \hat{\mathbf{z}} \\ \hat{\mathbf{z}}' &= \frac{1}{\sqrt{2}}(\hat{\mathbf{x}} + \hat{\mathbf{y}})\end{aligned}\tag{C.7}$$

and the matrix form is expressed in Fig. C.3. And finally for the (001) orientation we simply choose the z' axis to align normal to the surface which coincides with the z axis, while x' and y' coincide with x and y in the lab frame. To directly connect the cubic and lab coordinates, the compound matrix $K \equiv R^{-1}T$ is defined:

$$K_{(001)} = \begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix}\tag{C.8a}$$

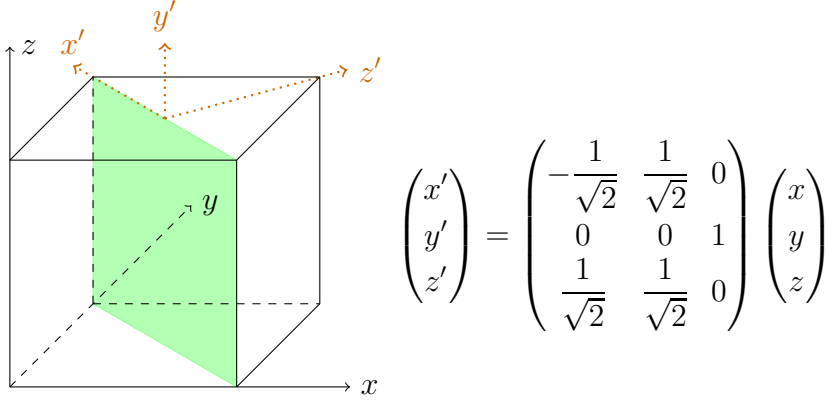


Figure C.3 The relation between (110) crystal “cut” (shaded) and the lab coordinates. The direction of the crystal orientation, \mathbf{z}' , is defined as the normal to the shaded surface, shown in dotted line as well as \mathbf{x}' axis, while \mathbf{y}' points straight up. The cubic coordinates are not show here.

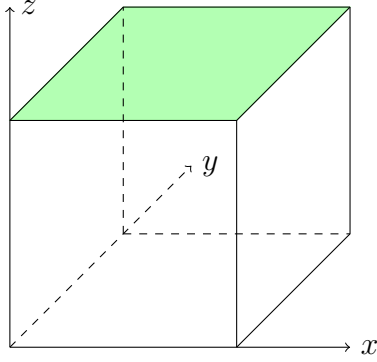
$$K_{(111)} = \begin{pmatrix} \sqrt{\frac{2}{3}} \cos \varphi & -\frac{\cos \varphi}{\sqrt{6}} - \frac{\sin \varphi}{\sqrt{2}} & -\frac{\cos \varphi}{\sqrt{6}} + \frac{\sin \varphi}{\sqrt{2}} \\ \sqrt{\frac{2}{3}} \sin \varphi & -\frac{\sin \varphi}{\sqrt{6}} + \frac{\cos \varphi}{\sqrt{2}} & -\frac{\sin \varphi}{\sqrt{6}} - \frac{\cos \varphi}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \end{pmatrix} \quad (\text{C.8b})$$

$$K_{(110)} = \begin{pmatrix} -\frac{\cos \varphi}{\sqrt{2}} & \frac{\cos \varphi}{\sqrt{2}} & -\sin \varphi \\ -\frac{\sin \varphi}{\sqrt{2}} & \frac{\sin \varphi}{\sqrt{2}} & -\cos \varphi \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{pmatrix} \quad (\text{C.8c})$$

The relation in the lab coordinate systems between the \mathbf{E} and the \mathbf{P} fields are achieved by three steps:

1) From \mathbf{E}_{xyz} to $\mathbf{E}_{\alpha\beta\gamma}$ by the rotation matrix:

$$\mathbf{E}_{\alpha\beta\gamma} = K^{-1} \mathbf{E}_{xyz} \quad (\text{C.9})$$



$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Figure C.4 The relation between (100) crystal “cut” (shaded) and the lab coordinates. The direction of the crystal orientation coincides with the axes of the lab frame thus renders an identity matrix.

2) From $\mathbf{E}_{\alpha\beta\gamma}$ to $\mathbf{P}_{\alpha\beta\gamma}$ by $\chi^{(3)}$.

3) From $\mathbf{P}_{\alpha\beta\gamma}$ to \mathbf{P}_{xyz} by the rotation matrix:

$$\mathbf{P}_{xyz} = K\mathbf{P}_{\alpha\beta\gamma} \quad (\text{C.10})$$

and it is used to derived the explicit form of tensor M .

$$\mathbf{P}_{\alpha\beta\gamma}^A \equiv \begin{bmatrix} P_{\alpha}^A \\ P_{\beta}^A \\ P_{\gamma}^A \end{bmatrix} = \begin{bmatrix} E_{\alpha}^a E_{\alpha}^b E_{\alpha}^c \\ E_{\beta}^a E_{\beta}^b E_{\beta}^c \\ E_{\gamma}^a E_{\gamma}^b E_{\gamma}^c \end{bmatrix} = \begin{bmatrix} (K' E_{xyz}^a)_{\alpha} (K' E_{xyz}^b)_{\alpha} (K' E_{xyz}^c)_{\alpha} \\ (K' E_{xyz}^a)_{\beta} (K' E_{xyz}^b)_{\beta} (K' E_{xyz}^c)_{\beta} \\ (K' E_{xyz}^a)_{\gamma} (K' E_{xyz}^b)_{\gamma} (K' E_{xyz}^c)_{\gamma} \end{bmatrix} \quad (\text{C.11})$$

and then in lab coordinates,

$$\begin{bmatrix} P_x^A \\ P_y^A \\ P_z^A \end{bmatrix} \equiv \mathbf{P}_{xyz}^A = K\mathbf{P}_{\alpha\beta\gamma}^A = K \begin{bmatrix} P_{\alpha}^A \\ P_{\beta}^A \\ P_{\gamma}^A \end{bmatrix} \\ = K \begin{bmatrix} (K'_{\alpha\alpha} E_x^a + K'_{\alpha\beta} E_y^a + K'_{\alpha\gamma} E_z^a)(K'_{\alpha\alpha} E_x^b + K'_{\alpha\beta} E_y^b + K'_{\alpha\gamma} E_z^b)(K'_{\alpha\alpha} E_x^c + K'_{\alpha\beta} E_y^c + K'_{\alpha\gamma} E_z^c) \\ (K'_{\beta\alpha} E_x^a + K'_{\beta\beta} E_y^a + K'_{\beta\gamma} E_z^a)(K'_{\beta\alpha} E_x^b + K'_{\beta\beta} E_y^b + K'_{\beta\gamma} E_z^b)(K'_{\beta\alpha} E_x^c + K'_{\beta\beta} E_y^c + K'_{\beta\gamma} E_z^c) \\ (K'_{\gamma\alpha} E_x^a + K'_{\gamma\beta} E_y^a + K'_{\gamma\gamma} E_z^a)(K'_{\gamma\alpha} E_x^b + K'_{\gamma\beta} E_y^b + K'_{\gamma\gamma} E_z^b)(K'_{\gamma\alpha} E_x^c + K'_{\gamma\beta} E_y^c + K'_{\gamma\gamma} E_z^c) \end{bmatrix}$$

$$\begin{aligned}
&= K \left[\begin{aligned}
&(K_{\alpha\alpha}E_x^a + K_{\beta\alpha}E_y^a + K_{\gamma\alpha}E_z^a)(K_{\alpha\alpha}E_x^b + K_{\beta\alpha}E_y^b + K_{\gamma\alpha}E_z^b)(K_{\alpha\alpha}E_x^c + K_{\beta\alpha}E_y^c + K_{\gamma\alpha}E_z^c) \\
&(K_{\alpha\beta}E_x^a + K_{\beta\beta}E_y^a + K_{\gamma\beta}E_z^a)(K_{\alpha\beta}E_x^b + K_{\beta\beta}E_y^b + K_{\gamma\beta}E_z^b)(K_{\alpha\beta}E_x^c + K_{\beta\beta}E_y^c + K_{\gamma\beta}E_z^c) \\
&(K_{\alpha\gamma}E_x^a + K_{\beta\gamma}E_y^a + K_{\gamma\gamma}E_z^a)(K_{\alpha\gamma}E_x^b + K_{\beta\gamma}E_y^b + K_{\gamma\gamma}E_z^b)(K_{\alpha\gamma}E_x^c + K_{\beta\gamma}E_y^c + K_{\gamma\gamma}E_z^c)
\end{aligned} \right] \\
&\hspace{20em} \text{(C.12)}
\end{aligned}$$

where again \mathbf{P}_{xyz}^A is the nonlinear polarization in *lab* frame and $\mathbf{P}_{\alpha\beta\gamma}^A$ is the nonlinear polarization in the *cubic* frame, with the superscript *A* representing the anisotropic part of the polarization only. So we see, for instance we can determine the elements of $M_{xxx}^x, M_{yyy}^x, M_{zzz}^x, \dots, M_{xyz}^x, \dots, M_{xyz}^z$ by

$$\begin{aligned}
P_x^A &= K_{\alpha\alpha}P_\alpha^A + K_{\alpha\beta}P_\beta^A + K_{\alpha\gamma}P_\gamma^A \\
&= (K_{\alpha\alpha}^4 + K_{\alpha\beta}^4 + K_{\alpha\gamma}^4)EEE|_{xxx}^{abc} \\
&\quad + (K_{\alpha\alpha}K_{\beta\alpha}^3 + K_{\alpha\beta}K_{\beta\beta}^3 + K_{\alpha\gamma}K_{\beta\gamma}^3)EEE|_{yyy}^{abc} \\
&\quad + (K_{\alpha\alpha}K_{\gamma\alpha}^3 + K_{\alpha\beta}K_{\gamma\beta}^3 + K_{\alpha\gamma}K_{\gamma\gamma}^3)EEE|_{zzz}^{abc} \\
&\quad + (K_{\alpha\alpha}^3K_{\beta\alpha} + K_{\alpha\beta}^3K_{\beta\beta} + K_{\alpha\gamma}^3K_{\beta\gamma})EEE|_{xxy,xyx,yxx}^{abc} \\
&\quad + (K_{\alpha\alpha}^3K_{\gamma\alpha} + K_{\alpha\beta}^3K_{\gamma\beta} + K_{\alpha\gamma}^3K_{\gamma\gamma})EEE|_{xzx,xzx,zxx}^{abc} \\
&\quad + (K_{\alpha\alpha}^2K_{\beta\alpha}^2 + K_{\alpha\beta}^2K_{\beta\beta}^2 + K_{\alpha\gamma}^2K_{\beta\gamma}^2)EEE|_{xyy,yyx,yyx}^{abc} \\
&\quad + (K_{\alpha\alpha}^2K_{\gamma\alpha}^2 + K_{\alpha\beta}^2K_{\gamma\beta}^2 + K_{\alpha\gamma}^2K_{\gamma\gamma}^2)EEE|_{xzz,zxz,zzx}^{abc} \\
&\quad + (K_{\alpha\alpha}K_{\beta\alpha}^2K_{\gamma\alpha} + K_{\alpha\beta}K_{\beta\beta}^2K_{\gamma\beta} + K_{\alpha\gamma}K_{\beta\gamma}^2K_{\gamma\gamma})EEE|_{yyz,yzy,zyy}^{abc} \\
&\quad + (K_{\alpha\alpha}K_{\beta\alpha}K_{\gamma\alpha}^2 + K_{\alpha\beta}K_{\beta\beta}K_{\gamma\beta}^2 + K_{\alpha\gamma}K_{\beta\gamma}K_{\gamma\gamma}^2)EEE|_{yzz,zyz,zzz}^{abc} \\
&\quad + (K_{\alpha\alpha}^2K_{\beta\alpha}K_{\gamma\alpha} + K_{\alpha\beta}^2K_{\beta\beta}K_{\gamma\beta} + K_{\alpha\gamma}^2K_{\beta\gamma}K_{\gamma\gamma})EEE|_{xyz,xzy,yxz,yzx,zxy,zyx}^{abc} \\
&\hspace{20em} \text{(C.13)}
\end{aligned}$$

where shorthand notations are used for the summations of terms with permutation subscripts, for example

$$EEE|_{xxz, xzx, zxx}^{abc} \equiv E_x^a E_x^b E_z^c + E_x^a E_z^b E_x^c + E_z^a E_x^b E_x^c \quad (\text{C.14})$$

The other two components of \mathbf{P}^A are written in the same way as

$$\begin{aligned} P_y^A &= K_{\beta\alpha} P_\alpha^A + K_{\beta\beta} P_\beta^A + K_{\beta\gamma} P_\gamma^A \\ &= (K_{\beta\alpha} K_{\alpha\alpha}^3 + K_{\beta\beta} K_{\alpha\beta}^3 + K_{\beta\gamma} K_{\alpha\gamma}^3) EEE|_{xxx}^{abc} \\ &\quad + (K_{\beta\alpha}^4 + K_{\beta\beta}^4 + K_{\beta\gamma}^4) EEE|_{yyy}^{abc} \\ &\quad + (K_{\beta\alpha} K_{\gamma\alpha}^3 + K_{\beta\beta} K_{\gamma\beta}^3 + K_{\beta\gamma} K_{\gamma\gamma}^3) EEE|_{zzz}^{abc} \\ &\quad + (K_{\alpha\alpha}^2 K_{\beta\alpha}^2 + K_{\alpha\beta}^2 K_{\beta\beta}^2 + K_{\alpha\gamma}^2 K_{\beta\gamma}^2) EEE|_{xxy, xyx, yxx}^{abc} \\ &\quad + (K_{\beta\alpha} K_{\alpha\alpha}^2 K_{\gamma\alpha} + K_{\beta\beta} K_{\alpha\beta}^2 K_{\gamma\beta} + K_{\beta\gamma} K_{\alpha\gamma}^2 K_{\gamma\gamma}) EEE|_{xxz, xzx, zxx}^{abc} \\ &\quad + (K_{\alpha\alpha} K_{\beta\alpha}^3 + K_{\alpha\beta} K_{\beta\beta}^3 + K_{\alpha\gamma} K_{\beta\gamma}^3) EEE|_{xyy, yxy, yyx}^{abc} \\ &\quad + (K_{\beta\alpha} K_{\alpha\alpha} K_{\gamma\alpha}^2 + K_{\alpha\beta} K_{\beta\beta} K_{\gamma\beta}^2 + K_{\beta\gamma} K_{\alpha\gamma} K_{\gamma\gamma}^2) EEE|_{xzz, zxx, zzx}^{abc} \\ &\quad + (K_{\beta\alpha}^3 K_{\gamma\alpha} + K_{\beta\beta}^3 K_{\gamma\beta} + K_{\beta\gamma}^3 K_{\gamma\gamma}) EEE|_{yyz, yzy, zyy}^{abc} \\ &\quad + (K_{\beta\alpha}^2 K_{\gamma\alpha}^2 + K_{\beta\beta}^2 K_{\gamma\beta}^2 + K_{\beta\gamma}^2 K_{\gamma\gamma}^2) EEE|_{yzz, zyz, zzy}^{abc} \\ &\quad + (K_{\alpha\alpha} K_{\beta\alpha}^2 K_{\gamma\alpha} + K_{\alpha\beta} K_{\beta\beta}^2 K_{\gamma\beta} + K_{\alpha\gamma} K_{\beta\gamma}^2 K_{\gamma\gamma}) EEE|_{xyz, xzy, yxz, yzx, zxy, zyx}^{abc} \end{aligned} \quad (\text{C.15})$$

$$\begin{aligned} P_z^A &= K_{\gamma\alpha} P_\alpha^A + K_{\gamma\beta} P_\beta^A + K_{\gamma\gamma} P_\gamma^A \\ &= (K_{\gamma\alpha} K_{\alpha\alpha}^3 + K_{\gamma\beta} K_{\alpha\beta}^3 + K_{\gamma\gamma} K_{\alpha\gamma}^3) EEE|_{xxx}^{abc} \end{aligned}$$

$$\begin{aligned}
& + (K_{\gamma\alpha}K_{\beta\alpha}^3 + K_{\gamma\beta}K_{\beta\beta}^3 + K_{\gamma\gamma}K_{\beta\gamma}^3)EEE|_{yyy}^{abc} \\
& + (K_{\gamma\alpha}^4 + K_{\gamma\beta}^4 + K_{\gamma\gamma}^4)EEE|_{zzz}^{abc} \\
& + (K_{\gamma\alpha}K_{\alpha\alpha}^2K_{\beta\alpha} + K_{\gamma\beta}K_{\alpha\beta}^2K_{\beta\beta} + K_{\gamma\gamma}K_{\alpha\gamma}^2K_{\beta\gamma})EEE|_{xxy,xyx,yxx}^{abc} \\
& + (K_{\alpha\alpha}^2K_{\gamma\alpha}^2 + K_{\alpha\beta}^2K_{\gamma\beta}^2 + K_{\alpha\gamma}^2K_{\gamma\gamma}^2)EEE|_{xxz,xzx,zzx}^{abc} \\
& + (K_{\gamma\alpha}K_{\alpha\alpha}K_{\beta\alpha}^2 + K_{\gamma\beta}K_{\alpha\beta}K_{\beta\beta}^2 + K_{\gamma\gamma}K_{\alpha\gamma}K_{\beta\gamma}^2)EEE|_{xyy,yxy,yyx}^{abc} \\
& + (K_{\alpha\alpha}K_{\gamma\alpha}^3 + K_{\alpha\beta}K_{\gamma\beta}^3 + K_{\alpha\gamma}K_{\gamma\gamma}^3)EEE|_{xzz,zxz,zzx}^{abc} \\
& + (K_{\beta\alpha}^2K_{\gamma\alpha}^2 + K_{\beta\beta}^2K_{\gamma\beta}^2 + K_{\beta\gamma}^2K_{\gamma\gamma}^2)EEE|_{yyz,yzy,zyy}^{abc} \\
& + (K_{\beta\alpha}K_{\gamma\alpha}^3 + K_{\beta\beta}K_{\gamma\beta}^3 + K_{\beta\gamma}K_{\gamma\gamma}^3)EEE|_{yzz,zyz,zzz}^{abc} \\
& + (K_{\alpha\alpha}K_{\beta\alpha}K_{\gamma\alpha}^2 + K_{\alpha\beta}K_{\beta\beta}K_{\gamma\beta}^2 + K_{\alpha\gamma}K_{\beta\gamma}K_{\gamma\gamma}^2)EEE|_{xyz,xzy,yxz,yzx,zxy,zyx}^{abc}
\end{aligned} \tag{C.16}$$

Now as a simple example for (100) symmetry axes, we can find the relation for the representations of E and P fields the lab frame. Let us firstly evaluate P_x^A (the shorthand notation for products of fields is used, such that $E_{xyy} \equiv E_x E_y E_y$):

$$\begin{aligned}
P_x^A &= P_\alpha^A \cos \varphi - P_\beta^A \sin \varphi \\
&= E_\alpha^a E_\alpha^b E_\alpha^c \cos \varphi - E_\beta^a E_\beta^b E_\beta^c \sin \varphi \\
&= (E_x^a \cos \varphi + E_y^a \sin \varphi)(E_x^b \cos \varphi + E_y^b \sin \varphi)(E_x^c \cos \varphi + E_y^c \sin \varphi) \cos \varphi \\
&\quad - (-E_x^a \sin \varphi + E_y^a \cos \varphi)(-E_x^b \sin \varphi + E_y^b \cos \varphi)(-E_x^c \sin \varphi + E_y^c \cos \varphi) \sin \varphi \\
&= E_x^a E_x^b E_x^c \cos^4 \varphi + E_x^a E_x^b E_y^c \cos^3 \varphi \sin \varphi + E_x^a E_y^b E_x^c \cos^3 \varphi \sin \varphi + E_x^a E_y^b E_y^c \cos^2 \varphi \sin^2 \varphi \\
&\quad + E_y^a E_x^b E_x^c \cos^3 \varphi \sin \varphi + E_y^a E_x^b E_y^c \cos^2 \varphi \sin^2 \varphi + E_y^a E_y^b E_x^c \cos^2 \varphi \sin^2 \varphi + E_y^a E_y^b E_y^c \cos \varphi \sin^3 \varphi
\end{aligned}$$

$$\begin{aligned}
& - [-E_x^a E_x^b E_x^c \sin^4 \varphi + E_x^a E_x^b E_y^c \cos \varphi \sin^3 \varphi + E_x^a E_y^b E_x^c \cos \varphi \sin^3 \varphi - E_x^a E_y^b E_y^c \cos^2 \varphi \sin^2 \varphi \\
& + E_y^a E_x^b E_x^c \cos \varphi \sin^3 \varphi - E_y^a E_x^b E_y^c \cos^2 \varphi \sin^2 \varphi - E_y^a E_y^b E_x^c \cos^2 \varphi \sin^2 \varphi \\
& + E_y^a E_y^b E_y^c \cos^3 \varphi \sin \varphi] \\
& = E_{xxx}^{abc} (\cos^4 \varphi + \sin^4 \varphi) + E_{xxy}^{abc} (\cos^3 \varphi \sin \varphi - \sin^3 \varphi \cos \varphi) + E_{xyx}^{abc} (\cos^3 \varphi \sin \varphi - \cos \varphi \sin^3 \varphi) \\
& + E_{xyy}^{abc} (\cos^2 \varphi \sin^2 \varphi + \cos^2 \varphi \sin^2 \varphi) \\
& + E_{yxx}^{abc} (\cos^3 \varphi + \cos \varphi \sin^3 \varphi) + E_{yxy}^{abc} (\cos^2 \varphi \sin^2 \varphi + \sin^2 \varphi \cos^2 \varphi) \\
& + E_{yyx}^{abc} (\cos^2 \varphi \sin^2 \varphi - \cos^2 \varphi \sin^2 \varphi) \\
& + E_{yyy}^{abc} (\cos \varphi \sin^3 \varphi - \cos^3 \varphi \sin \varphi) \tag{C.17}
\end{aligned}$$

then we get the matrix elements for M in (100) crystal face,

$$x, xxx (P_x^A \leftarrow E_{xxx}) : \cos^4 \varphi + \sin^4 \varphi = \frac{1}{4} (3 + \cos 4\varphi) \tag{C.18}$$

$$x, yyy (P_x^A \leftarrow E_{yyy}) : \cos \varphi \sin^3 \varphi - \cos^3 \varphi \sin \varphi = -\frac{1}{4} \sin 4\varphi \tag{C.19}$$

and they agree with the expression derived from the general discussion:

$$M_{xxx}^x = K_{\alpha\alpha}^4 + K_{\alpha\beta}^4 + K_{\alpha\gamma}^4 = \cos^4 \varphi + \sin^4 \varphi = \frac{1}{4} (3 + \cos 4\varphi) \tag{C.20}$$

$$M_{yyy}^x = K_{\alpha\alpha} K_{\beta\alpha}^3 + K_{\alpha\beta} K_{\beta\beta}^3 + K_{\alpha\gamma} K_{\beta\gamma}^3 = \cos \varphi \sin^3 \varphi + (-\sin \varphi) \cos^3 \varphi + 0 = -\frac{1}{4} \sin 4\varphi \tag{C.21}$$

Similarly, as another example, for (111) crystal symmetry axes,

$$\begin{aligned}
M_{xxz}^x & = K_{\alpha\alpha}^3 K_{\gamma\alpha} + K_{\alpha\beta}^3 K_{\gamma\beta} + K_{\alpha\gamma}^3 K_{\gamma\gamma} \\
& = \left(\sqrt{\frac{2}{3}} \cos \varphi \right)^3 \frac{1}{\sqrt{3}} + \left(-\frac{\cos \varphi}{\sqrt{6}} - \frac{\sin \varphi}{\sqrt{2}} \right)^3 \frac{1}{\sqrt{3}} + \left(-\frac{\cos \varphi}{\sqrt{6}} + \frac{\sin \varphi}{\sqrt{2}} \right)^3 \frac{1}{\sqrt{3}}
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{\sqrt{3}} \left[\frac{2}{3} \sqrt{\frac{2}{3}} \cos^3 \varphi + 2 \left(-\frac{\cos \varphi}{\sqrt{6}} \right)^3 + 6 \left(-\frac{\cos \varphi}{\sqrt{6}} \right) \left(\frac{\sin \varphi}{\sqrt{2}} \right)^2 \right] \\
&= \frac{1}{\sqrt{3}} \left[\frac{4}{\sqrt{6}} \cos^3 \varphi - \frac{3}{\sqrt{6}} \cos \varphi \right] \\
&= \frac{1}{\sqrt{18}} \cos 3\varphi
\end{aligned} \tag{C.22}$$

And finally for (110) axes,

$$\begin{aligned}
M_{xxx}^x &= K_{\alpha\alpha}^4 + K_{\alpha\beta}^4 + K_{\alpha\gamma}^4 \\
&= \left(-\frac{\cos \varphi}{\sqrt{2}} \right)^4 + \left(\frac{\cos \varphi}{\sqrt{2}} \right)^4 + (-\sin \varphi)^4 \\
&= \frac{1}{2} \cos^4 \varphi + \sin^4 \varphi \\
&= \frac{3}{16} \cos 4\varphi + \frac{1}{16} (9 - 4 \cos 2\varphi)
\end{aligned} \tag{C.23}$$

$$\begin{aligned}
M_{zzz}^z &= K_{\gamma\alpha}^4 + K_{\gamma\beta}^4 + K_{\gamma\gamma}^4 \\
&= \left(\frac{1}{\sqrt{2}} \right)^4 + \left(\frac{1}{\sqrt{2}} \right)^4 + 0^4 \\
&= \frac{1}{2}
\end{aligned} \tag{C.24}$$

$$\begin{aligned}
M_{xxy}^y &= K_{\alpha\alpha}^2 K_{\beta\alpha}^2 + K_{\alpha\beta}^2 K_{\beta\beta}^2 + K_{\alpha\gamma}^2 K_{\beta\gamma}^2 \\
&= \left(-\frac{\cos \varphi}{\sqrt{2}} \right)^2 \left(-\frac{\sin \varphi}{\sqrt{2}} \right)^2 + \left(\frac{\cos \varphi}{\sqrt{2}} \right)^2 \left(\frac{\sin \varphi}{\sqrt{2}} \right)^2 + (-\sin \varphi)^2 (\cos \varphi)^2 \\
&= \frac{3}{16} (1 - \cos 4\varphi)
\end{aligned} \tag{C.25}$$

$$\begin{aligned}
M_{xxx}^y &= K_{\beta\alpha} K_{\alpha\alpha}^3 + K_{\beta\beta} K_{\alpha\beta}^3 + K_{\beta\gamma} K_{\alpha\gamma}^3 \\
&= \left(-\frac{\sin \varphi}{\sqrt{2}} \right) \left(-\frac{\cos \varphi}{\sqrt{2}} \right)^3 + \left(\frac{\sin \varphi}{\sqrt{2}} \right) \left(\frac{\cos \varphi}{\sqrt{2}} \right)^3 + \cos \varphi (-\sin \varphi)^3 \\
&= \frac{\sin \varphi \cos^3 \varphi}{2} - \sin^3 \varphi \cos \varphi \\
&= \frac{3}{16} \sin 4\varphi - \frac{1}{8} \sin 2\varphi
\end{aligned} \tag{C.26}$$

Finally the tensor M , representing the cubic rotation relative to the lab coordinates, is determined for different forms for different cubic symmetry axe choices, i.e.

$\langle 100 \rangle, \langle 111 \rangle, \langle 110 \rangle$

$$\langle 100 \rangle \tag{C.27}$$

$$M_{jkl}^i \begin{array}{ccc} & \begin{array}{c} xxx \\ 3 + \cos 4\varphi \\ 4 \end{array} & \begin{array}{c} yyy \\ -\frac{3 + \sin 4\varphi}{4} \end{array} & \begin{array}{c} zzz \\ 0 \end{array} \\ \begin{array}{c} x \\ y \\ z \end{array} & & & \end{array}$$

$$\begin{array}{l} (continued) \\ M_{jkl}^i \end{array} \begin{array}{ccccccc} & \begin{array}{c} xxy \\ \frac{\sin 4\varphi}{4} \end{array} & \begin{array}{c} xxz \\ 0 \end{array} & \begin{array}{c} xyx \\ \frac{1 - \cos 4\varphi}{4} \end{array} & \begin{array}{c} xzz \\ 0 \end{array} & \begin{array}{c} yyz \\ 0 \end{array} & \begin{array}{c} yzx \\ 0 \end{array} & \begin{array}{c} xyz \\ 0 \end{array} \\ \begin{array}{c} x \\ y \\ z \end{array} & & & & & & & \end{array}$$

$$\langle 111 \rangle \tag{C.28}$$

$$M_{jkl}^i \begin{array}{ccc} & \begin{array}{c} xxx \\ \frac{1}{2} \end{array} & \begin{array}{c} yyy \\ 0 \end{array} & \begin{array}{c} zzz \\ 0 \end{array} \\ \begin{array}{c} x \\ y \\ z \end{array} & & & \end{array}$$

$$\begin{array}{l} (continued) \\ M_{jkl}^i \end{array} \begin{array}{ccccccc} & \begin{array}{c} xxy \\ 0 \end{array} & \begin{array}{c} xxz \\ \frac{\cos 3\varphi}{\sqrt{18}} \end{array} & \begin{array}{c} xyx \\ \frac{1}{6} \end{array} & \begin{array}{c} xzz \\ \frac{1}{3} \end{array} & \begin{array}{c} yyz \\ -\frac{\cos 3\varphi}{\sqrt{18}} \end{array} & \begin{array}{c} yzx \\ 0 \end{array} & \begin{array}{c} xyz \\ \frac{\sin 3\varphi}{\sqrt{18}} \end{array} \\ \begin{array}{c} x \\ y \\ z \end{array} & & & & & & & \end{array}$$

$$\langle 110 \rangle \tag{C.29}$$

$$\begin{array}{c}
M_{jkl}^i \\
x \\
y \\
z
\end{array}
\begin{array}{ccc}
& xxx & yyy & zzz \\
\frac{9 - \cos 2\varphi + 3 \cos 4\varphi}{16} & \frac{-2 \sin 2\varphi - 3 \sin 4\varphi}{16} & & \\
\frac{-2 \sin 2\varphi + 3 \sin 4\varphi}{16} & \frac{9 + 4 \cos 2\varphi + 3 \cos 4\varphi}{16} & & \\
0 & 0 & & \frac{1}{2}
\end{array}$$

$$\begin{array}{c}
M_{jkl}^i \\
x \\
y \\
z
\end{array}
\begin{array}{ccc}
& xxy & xxz & xyy \\
\frac{-2 \sin 2\varphi + 3 \sin 4\varphi}{16} & 0 & \frac{3(1 - \cos 4\varphi)}{16} & \\
\frac{3(1 - \cos 4\varphi)}{16} & 0 & \frac{-2 \sin 2\varphi - 3 \sin 4\varphi}{16} & \\
0 & \frac{1 + \cos 2\varphi}{4} & 0 &
\end{array}$$

$$\begin{array}{c}
M_{jkl}^i \\
x \\
y \\
z
\end{array}
\begin{array}{cccc}
& xzz & yyz & yzz & xyz \\
\frac{1 + \cos 2\varphi}{4} & 0 & \frac{\sin 2\varphi}{4} & 0 & \\
\frac{\sin 2\varphi}{4} & 0 & \frac{1 - \cos 2\varphi}{4} & 0 & \\
0 & \frac{1 - \cos 2\varphi}{4} & 0 & \frac{\sin 2\varphi}{4} &
\end{array}$$

For the special angle value $\varphi = 0$, M takes simple forms

$$\langle 100 \rangle$$

$$\begin{array}{c}
M_{jkl}^i \\
x \\
y \\
z
\end{array}
\begin{array}{cccccc}
xxx & yyy & zzz & xxy & xxz & xyy \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0
\end{array}
\tag{C.30}$$

$\langle 111 \rangle$

$$\begin{array}{cccccccccccc}
M_{jkl}^i & xxx & yyy & zzz & xxy & xxz & xyx & xzz & yyz & yzz & xyz \\
x & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{\sqrt{18}} & \frac{1}{6} & \frac{1}{3} & -\frac{1}{\sqrt{18}} & 0 & 0 \\
y & 0 & \frac{1}{2} & 0 & \frac{1}{6} & 0 & 0 & 0 & 0 & \frac{1}{3} & -\frac{1}{\sqrt{18}} \\
z & \frac{1}{\sqrt{18}} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & -\frac{1}{\sqrt{18}} & 0 & \frac{1}{3} & 0 & 0
\end{array} \tag{C.31}$$

$\langle 110 \rangle$

$$\begin{array}{cccccccccccc}
M_{jkl}^i & xxx & yyy & zzz & xxy & xxz & xyx & xzz & yyz & yzz & xyz \\
x & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\
y & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
z & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0
\end{array} \tag{C.32}$$

where the simplest case for (100) surface placed crystal with no rotation, the tensor

M takes the form:

$$M_{jkl}^i = \begin{cases} 1 & \text{when } jkl = xxx, yyy \text{ or } zzz \\ 0 & \text{otherwise} \end{cases} \tag{C.33}$$

where $i = x, y$ or z . So the anisotropic term in (III.18) is simplified to

$$\chi_d M :: \mathbf{E}^a \mathbf{E}^b \mathbf{E}^c = \chi_d (\hat{\mathbf{x}} E_x^a E_x^b E_x^c + \hat{\mathbf{y}} E_y^a E_y^b E_y^c + \hat{\mathbf{z}} E_z^a E_z^b E_z^c) \tag{C.34}$$

and we are going to use this simplest configuration in (C.33) or (C.34) to illustrate the nonlinear SPP properties, without the cumbersome geometry distraction, in all the related chapters.

APPENDIX D

NUMERICAL ALGORITHM FOR NLS EQUATIONS

The NLS equation with a loss term (V.55) is prototyped in the class of convection-diffusion equations for the Finite difference method for partial differential equations [110]. For both accuracy and efficiency in computation, the implicit Crank-Nicolson scheme is adapted for the NLS equation [111]. For notational convenience and comparison to heat equation, the NLS equation (V.55) is rewritten with space-time variables switched, with first derivative on the left side and other terms on the right side:

$$i \frac{\partial u}{\partial t} = -\frac{1}{2} \frac{\partial^2 u}{\partial x^2} = |u|^2 u + i\Gamma u \quad (\text{D.1})$$

To discretize the equation, we use forward-time and central-space meshes and label the time-mesh interval Δt and the space-mesh interval Δx . For example, the time derivative term on the left side of (D.1) is

$$\frac{\partial u}{\partial t} \simeq \frac{u_k^{n+1} - u_k^n}{\Delta t} \quad (\text{D.2})$$

where the superscripts $n + 1$ and n label the time step with $n(= 1, \dots, N)$ as the current time step, and the subscript $k(= 1, \dots, K)$ labels the central space point. The last term on the right hand side of (D.1) is represented by its central space

point, but average of forward time steps:

$$u \simeq \frac{u_k^{n+1} + u_k^n}{2} \quad (\text{D.3})$$

and similarly the second term:

$$|u|^3 \simeq \frac{|u_k^{n+1}|^3 + |u_k^n|^3}{2} \quad (\text{D.4})$$

As for the second spatial derivative term $\frac{\partial^2 u}{\partial x^2}$, we firstly write the standard central-space derivative at current time step n :

$$\frac{\partial^2 u}{\partial x^2} \simeq \frac{u_{k+1}^n - u_k^n + u_{k-1}^n}{\Delta x^2} \quad (\text{D.5})$$

then replace every term u^n by its forward time average $(u^n + u^{n+1})/2$:

$$\frac{\partial^2 u}{\partial x^2} \simeq \frac{u_{k+1}^n - u_k^n + u_{k-1}^n}{2\Delta x^2} + \frac{u_{k+1}^{n+1} - u_k^{n+1} + u_{k-1}^{n+1}}{2\Delta x^2} \quad (\text{D.6})$$

Define the diffusion number, which is dimensionless and positive

$$\lambda = \frac{\Delta t}{\Delta x^2} \quad (\text{D.7})$$

and collect all discrete expression for (D.1), we have now

$$\begin{aligned} 2i(u_k^{n+1} - u_k^n) &= -\frac{\lambda}{2}(u_{k+1}^{n+1} - u_k^{n+1} + u_{k-1}^{n+1}) + \Delta t (|u_k^{n+1}|^3 + i\Gamma u_k^{n+1}) \\ &\quad -\frac{\lambda}{2}(u_{k+1}^n - u_k^n + u_{k-1}^n) + \Delta t (|u_k^n|^3 + i\Gamma u_k^n) \end{aligned} \quad (\text{D.8})$$

Group the linear forward time steps $n + 1$ to one side, while the current time steps n and higher orders of $n + 1$ to the other side:

$$\begin{aligned} &\frac{\lambda}{2}u_{k-1}^{n+1} + \left[2i - \left(\frac{\lambda}{2} + i\Gamma\Delta t\right)\right]u_k^{n+1} + \frac{\lambda}{2}u_{k-1}^{n+1} \\ &= -\frac{\lambda}{2}u_{k-1}^n + \left[2i + \left(\frac{\lambda}{2} + i\Gamma\Delta t\right)\right]u_k^n - \frac{\lambda}{2}u_{k-1}^n + \Delta t (|u_k^{n+1}|^3 + |u_k^n|^3) \end{aligned} \quad (\text{D.9})$$

To update all spatial points $k = 1, \dots, K$ for time step $n + 1$, the previous time step n values are used, as illustrated in Fig. D.1. Define $2\Lambda \equiv \frac{\lambda}{2} + i\Gamma\Delta t$ and use

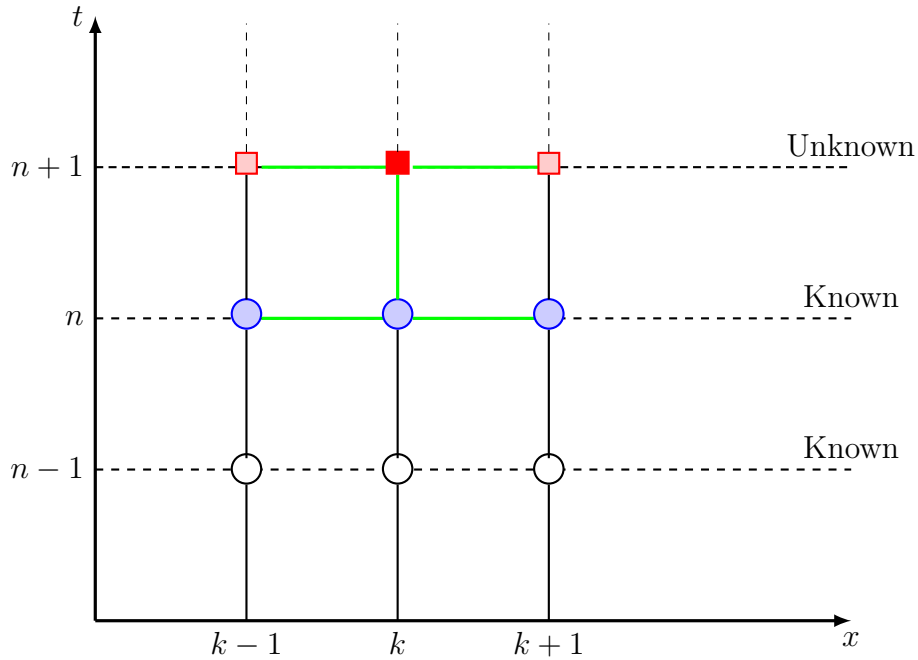


Figure D.1 Crank-Nicolson scheme illustrated with every point representing a value of u_k^n at time step n and position k . The circles are grid points that are known at current time step and the squares are unknown points. The forward time step point, indicated as a solid square (red), is determined by known points with previous time step, indicated as shaded circles (blue), and unknown grid points, indicated as hollow squares (red), at the same time step using an implicit method.

the matrix form:

$$\begin{aligned}
& \begin{pmatrix} 2(i-\Lambda) & \frac{\lambda}{2} & 0 & \cdots & 0 & 0 & -\frac{\lambda}{2} \\ \frac{\lambda}{2} & 2(i-\Lambda) & \frac{\lambda}{2} & \cdots & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & & & 0 \\ 0 & \cdots & \frac{\lambda}{2} & 2(i-\Lambda) & \frac{\lambda}{2} & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & & 0 \\ 0 & 0 & \cdots & 0 & \frac{\lambda}{2} & 2(i-\Lambda) & \frac{\lambda}{2} \\ -\frac{\lambda}{2} & 0 & \cdots & 0 & 0 & \frac{\lambda}{2} & 2(i-\Lambda) \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_{k-1} \\ u_k \\ u_{k+1} \\ \vdots \\ u_K \end{pmatrix}^{n+1} \\
= & \begin{pmatrix} 2(i+\Lambda) & -\frac{\lambda}{2} & 0 & \cdots & 0 & 0 & \frac{\lambda}{2} \\ -\frac{\lambda}{2} & 2(i+\Lambda) & -\frac{\lambda}{2} & \cdots & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & & & 0 \\ 0 & \cdots & -\frac{\lambda}{2} & 2(i+\Lambda) & -\frac{\lambda}{2} & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & & 0 \\ 0 & 0 & \cdots & 0 & -\frac{\lambda}{2} & 2(i+\Lambda) & -\frac{\lambda}{2} \\ \frac{\lambda}{2} & 0 & \cdots & 0 & 0 & -\frac{\lambda}{2} & 2(i+\Lambda) \end{pmatrix} \begin{pmatrix} u_1 \\ \vdots \\ u_{k-1} \\ u_k \\ u_{k+1} \\ \vdots \\ u_K \end{pmatrix}^n \\
& + \Delta t \left[\begin{pmatrix} |u_1|^3 \\ \vdots \\ |u_{k-1}|^3 \\ |u_k|^3 \\ |u_{k+1}|^3 \\ \vdots \\ |u_K|^3 \end{pmatrix}^n + \begin{pmatrix} |u_1|^3 \\ \vdots \\ |u_{k-1}|^3 \\ |u_k|^3 \\ |u_{k+1}|^3 \\ \vdots \\ |u_K|^3 \end{pmatrix}^{n+1} \right]
\end{aligned} \tag{D.10}$$

where the common time step stamps n and $n+1$ are labeled out side of the column vectors in the matrix expression. Accordingly we can define matrix A , column

vector U and matrix B to simplify the notation such that (D.10) is written as

$$AU^{n+1} = BU^n + \Delta t (|U^{n+1}|^3 + |U^n|^3) \quad (\text{D.11})$$

where the superscripts n and $n + 1$ label time steps while 3 means to the third power. Now the U^{n+1} update formula is

$$U^{n+1} = A^{-1}BU^n + A^{-1}\Delta t|U^n|^3 + A^{-1}\Delta t|U^{n+1}|^3 \quad (\text{D.12})$$

The right side of the update scheme, however, contains a forward time term $|U^{n+1}|^3$. Therefore, an implicit iteration method is used to determine the new time step value for U^{n+1} . Namely, to achieve the next time step U^{n+1} , we start with a current time step value U^n for all U terms on the right side and get a tentative value $U^{n+1}(i = 0)$ on the left side for the $i = 0$ iteration. Then the $U^{n+1}(i = 0)$ value is used back on the right side to get an improved result for the left side $U^{n+1}(i = 1)$. Keep the iteration to improve the forward time U^{n+1} term until $i = \infty$ or the accuracy is within a small range of tolerance.

The expected solution is a dark soliton and it is a hole imbedded in a non-zero background. Thus the boundary condition for the numerical method, instead of commonly used periodic boundary condition [110] or transparent boundary condition [112], an inverse-periodic boundary condition is used, which has a π phase shift between the left end and right end of the grid. This inverse-periodic boundary condition is represented in the matrix A and B by the off-diagonal corner

Table D.1 Numerical solution with lossless metals.

$c = 3 \times 10^8(m/s)$ $\varepsilon_0 = 8.8542 \times 10^{-12}(F/m)$ $\varepsilon_d = 1$	Speed of light in vacuum Vacuum permittivity Dielectric constant of vacuum
$\omega_p = 1.3814 \times 10^{16}(rad/s)$ $\varepsilon_\infty = 5.1$	Plasma frequency of metal Background dielectric of metal
$\lambda_0 = 1500(nm)$ $k_0 = \frac{2\pi}{\lambda_0} = 4.18879 \times 10^6(rad/m)$ $\omega_0 = \frac{2\pi c}{\lambda_0} = 1.25664 \times 10^{15}(rad/s)$ $\beta_0 = \beta(\omega_0) = 4.21013 \times 10^6(rad/m)$ $\beta_1 = \beta'(\omega_0) = 3.38631 \times 10^{-9}(s/m)$ $\beta_2 = \beta''(\omega_0) = 9.27626 \times 10^{-26}(s^2/m)$	Laser wavelength Laser wavevector in vacuum Angular frequency of laser SPP wavevector($\sim k_{spp}$) Inverse of SPP group velocity ($1/v_g$) SPP GVD
$\chi = \chi^{(3)} = 2.8 \times 10^{-19}(m^2/V^2)$ $\nu = \frac{k_0 \chi^{(3)} \sqrt{\varepsilon_d^3}}{\sqrt{16 \varepsilon_{m0} (\varepsilon_{m0} - \varepsilon_d)^3}}$ $= 2.95 \times 10^{-17}(m/V^2)$	Nonlinear susceptibility of metal Nonlinear SPP wavevector variant
Area= $0.01(mm^2)$ $\tau_0 = 10(ps)$ $L_0 = \tau_0/\beta_1 = 2.95 \times 10^{-3}(m)$ $L_D = \tau_0^2/\beta_2 = 1078(m)$ $f_{rep} = 80(MHz)$ $A_0 = \frac{\sqrt{\beta_2}}{\sqrt{\nu}\tau_0 \cos \phi} = 5.6 \times 10^6(V/m)$ $P_{ave} = 0.338(W)$	Laser spot size Laser pulse duration Pulse length, 2.95mm SPP Dispersion length Pulsed laser repetition value Threshold amplitude for fundamental soliton Average power (pulsed laser)

elements, which possess an additional negative sign comparing to their "neighbor" elements.

The parameters used in the numerical calculation and the results are tabulated in Tab.D.1 and Tab.D.2

Table D.2 Numerical solution with lossy metals.

$\omega_p = 1.3814 \times 10^{16}(\text{rad}/s)$ $\gamma_p = 6.857 \times 10^{13}(\text{rad}/s)$ $\varepsilon_\infty = 5.1$	Plasma frequency of silver Damping coefficient Background dielectric of metal
$\lambda_0 = 1500(\text{nm})$ $\tau_0 = 8(\text{fs})$ $L_0 = \tau_0/\beta_1 = 2.36(\mu\text{m})$ $L_{spp} = 641(\mu\text{m})$ $L_D = 741(\mu\text{m})$ $L_{NL} = \frac{1}{\text{Re}\nu A ^2} = 679(\mu\text{m})$ $f_{rep} = 1(\text{Hz})$ $A_0 = \frac{\sqrt{\beta_2}}{\sqrt{\nu}\tau_0 \cos \phi} = 7420(\text{V}/\text{m})$ $P_{ave} = 5.77(\mu\text{W})$ $\Gamma = 2.31$	Laser wavelength Laser pulse duration Pulse length SPP propagation length SPP Dispersion length Nonlinear effect length Pulsed laser repetition value Threshold amplitude for fundamental soliton Pulse power Loss coefficient

```

%MatLab source code;
%Source Code for Crank Nicolson Method – Crank Nicolson.m
clear all;
clc;
%clf;
%%
%Parameters
mesh_size = 1000;
total_step = 2000;
dx = 0.02;
dt = dx * dx / 5.0;
amplitude = 4.0;
%velocity = 0.10;
grid_mem = 3;
Gamma = -0.0;%damping
%%
lambda=dt/dx/dx;
    
```

```

diag_ele = lambda - li*dt*Gamma;
off_ele = -0.5*lambda;
% diagnal      -1sub  +1super
i=[1:mesh_size 2:mesh_size 1:mesh_size-1];
j=[1:mesh_size 1:mesh_size-1 2:mesh_size];%(i,j)->s
s=[diag_ele*ones(1,mesh_size) off_ele*ones(1,mesh_size-1)...
off_ele*ones(1,mesh_size-1)];
imagi = li*speye(mesh_size);
major = sparse(i,j,s);
leftA = imagi + major;
leftA(1,mesh_size) = -off_ele;
leftA(mesh_size,1) = -off_ele;
rightB = imagi - major;
rightB(1,mesh_size) = off_ele;
rightB(mesh_size,1) = off_ele;
clear diag_ele off_ele i j s;
invA=inv(leftA);
%%
%Graph Options
n_lines = 10;
line_skip = max(1, uint32(total_step / n_lines));
n_points = 100;
point_skip = max(1, mesh_size / n_points);
%%
%Initial Condition
x = (-mesh_size/2+1:mesh_size/2) *dx;
init_line = amplitude*tanh(x);
%init_line = -2i*amplitude*exp(-2i*velocity*x).*sech(2*amplitude*x);
%%
%Grid prepare
grids = [init_line; init_line; init_line];
%
```

```

for time_step = grid_mem: 2*grid_mem

    t3 = mod(time_step - 1, grid_mem) + 1;%3
    t2 = mod(time_step - 2, grid_mem) + 1;%2
    t1 = mod(time_step - 3, grid_mem) + 1;%1

%FTCS

    u2C = double(grids(t2, :));%Center
    u2L = double([grids(t2,1) grids(t2,1:end-1)]);%Left abrupt boundary condition
    u2R = double([grids(t2,2:end) grids(t2,end)]);%Right
    grids(t3,:) = double(1i*lambda*(2*u2C-(u2L+u2R)) + ...
        (1+2i*dt*u2C.* conj(u2C)).*u2C);
end

clear t3 t2 t1 u2C u2L u2R

water = abs(grids(grid_mem,1:point_skip:mesh_size));

%%
%evolution
figure(1);
clf;
plot(x, abs(grids(grid_mem,:)))
hold on;

for time_step = grid_mem:total_step

    t3 = mod(time_step - 1, grid_mem) + 1;%3 today
    t2 = mod(time_step - 2, grid_mem) + 1;%2 yesterday
    t1 = mod(time_step - 3, grid_mem) + 1;%1 the day before yesterday
    u2 = grids(t2, :);

%Du_Fort
%     alpha = double(1i/2/dt + 1/dx/dx);
%     beta = double(1i/2/dt - 1/dx/dx);
%     u2L = double([grids(t2,1) grids(t2,1:end-1)]);%Left -
%     u2R = double([grids(t2,2:end) grids(t2,end)]);%Right
%     u2C = grids(t2, :);
%     u1C = double(grids(t1, :));%Center
%     grids(t3,:) = double((beta*u1C + (u2L+u2R)/dx/dx - ...

```



```

%2*u2C.* conj(u2C).*u2C )/alpha) +1i*Gamma*u2C/alpha;
%
%Crank_Nicolson
    rightF1 = u2*rightB.';
    rightF2 = 0*dt.*u2.*conj(u2).*u2;
    rightFixed = rightF1+rightF2;
%    vnewU=[u2;u2];%dimension reservation
%    toler = 1e-12;%tolerance
%    iter_old = 1;
%    iter_new = 2;
%    err_t = 1;
%    while( err_t > toler)
%        oldU = vnewU(iter_old,:);
%        rhside = rightFixed + dt*oldU.*conj(oldU).*oldU;
%        vnewU(iter_new,:) = invA*rhside';
%        err_t = max(abs((vnewU(1,:)-vnewU(2,:))));
%        iter_new = 3 - iter_new;
%        iter_old = 3 - iter_old;
%    end
%    grids(t3,:) = vnewU(iter_new,:);

%    for iter = 1:3
%        rhside = rightFixed + 0*dt*u2.*conj(u2).*u2;
%        u2 = rhside*invA.';
%    end
grids(t3,:) = u2;

if(mod(time_step, line_skip) == 0)
    disp(time_step);
    figure(1);
    plot(x, abs(grids(t3,:)));
    water = [water;abs(grids(t3,1:point_skip:mesh_size))];

```

```
        end
    end
    figure(2);
    waterfall(water);
    hold off;
%end
```

APPENDIX E

C/C++ SOURCE CODE FOR FDTD METHOD

FDTDmain.cpp

```
#include <cstdlib>
#include <iostream>

// This program is a simulation for SPP propagation on a metal surface.
// The simulation for linear metal is stable
// The simulation for nonlinear metal is not stable for large nonlinear
// susceptibilities, because metal has negative dielectric constant
// which does not compatible with a Kerr nonlinear index.

//Reference Book:Allen Taflovve, Susan C. Hagness, Computational electrodynamics,
//The finite-difference time-domain method, 3rd edition, page 203

//This program is for: Complex vs Real calculation optional.

//This program is ALL real valued. Complex numbers are computed by separating
//real and imaginary parts
//Unit:D=H*k/omega
//unit: SI (modified)
#include "math.h"
#include "stdio.h"
#include "stdlib.h"
#include "algorithm"
```

```

#include "time.h"
#include "FDTDarray.h"
#include "FDTDcubic.h"
#include "FDTDtime.h"
#define SQR2 1.414216562373//sqrt(2)
#define PI 3.1415926535898
using namespace std;
// parameters definition, caution: only constants are defined as extern variables
/*
Naming convention notes:
    1.upper case for grids values
    2."_0" for constants
    3."_func" for functions
    4."_r" for relative, dimensionless values
    5.prefix "d" means difference
    6.prefix "C" for factors
    7.postscript "c" for complex values
*/
//
/*-----Universal Constants-----*/
double const epsilon_0=8.854187817e-3;
//unit: nF/m,dielectric constant in vacuum. 1E-12-->F/m
double const mu_0=0.4e3*PI;
//unit: nH/m, permeability constant in vacuum. 1E-6-->H/m
double const c_0=1/sqrt(epsilon_0*mu_0);
//==0.29,unit: nm/attosec, speed of light in vacuum. 1E-9/1E-18=1E9-->m/s
double lambda_0=800;
//unit:nm, center optical wavelength. 1E-9-->m
double const omega_0=2*PI*c_0/lambda_0;
//unit:rad/attosec, central angular frequency,(1E9*1E9=1E18-->Hz);
double const k_0=omega_0/c_0;
//unit: /nm,wavevector. 1E9-->/m
// typical optical (500nm) ~ 2PI(0.0006) GGHz=2PI(6.((E-4))) GGHz -->2PI(6.E14)Hz

```

```

double const epsilon_inf=5.1;
//for silver,use relative values!
//-----For Debye Model (Optional)-----
double const epsilon_sp=1;
//-16643.2*epsilon_0;//pF/m.static response, -16643.2 for silver
// double const epsilon_inf=1;//5.1*epsilon_0;//infinite response, 10.9866 for silver
double const depsilon_p=0;
//epsilon_sp-epsilon_inf;//difference
double const tau_p=9076.707;
//relaxation time:attosec;9.076707e-15 s for silver
//-----For Lorentz Model (Optional)-----
double const delta_p=0;
//1E-4*omega_p;//damping coefficient,TBA
//-----For Drude Model (in use)-----
double const omega_p=1.3814E-2;
//frequency of pole pair,GGHz
double const gamma_p=3.18786E-6;
//relaxation time,unit GGHz
//-----Kerr Nonlinearity (optional)-----
//ref. Nonlinear Optics by Rober W. Boyd, 3rd Edition, Page 222,212
// double const N_atom=4e22;//(cm)-3 atom number density
// double const e_charge=1.6e-19;//Coulomb, unit charge
// double const m_atom=1;
double const alpha=1;//alpha=1 for All Kerr and None Raman
double const chi3M=2.8e-19;//2.8e2;//2.8;
//unit(m2/V2)-----Ag:2.8e-19 (m2/V2),Au 7.6e-19
//-----Raman Effect (Optional)-----
//ref. Nonlinear Optics by Rober W. Boyd, 3rd Edition, Page 222
double const a0_bohr=0;//0.05;//unit: nm,Bohr radius,
double const v_electron=0;//c_0/137;//nm/attosec,typical electronic velocity
double const tau1=0;
//2*PI*a0_bohr/v_electron;//143attosec, or 1E-16 second;

```

```

//a single Lorentzian line centered on the optical phonon frequency 1/t1,TBA
double const tau2=0;//tau1/1E4;
//and having a bandwidth of 1/t2,the reciprocal phonon lifetime ,TBA
double const omega_raman=0;//sqrt(tau1*tau1+tau2*tau2)/tau1/tau2;
double const delta_raman=0;//1/tau2;
//-----SPP parameters-----
double const drudedenom=(pow(omega_0,4)-pow(omega_0*gamma_p,2));
double const epsilon_R=epsilon_inf-pow(omega_p*omega_0,2)/drudedenom;
//real part of dielectric constant
double const epsilon_I=omega_0*omega_p*omega_p*gamma_p/drudedenom;
//imaginary part of dielectric constant
double const epsilon0R2=(1+epsilon_R)*(1+epsilon_R);//squared
double const kx_R=k_0*sqrtReal(epsilon_R,epsilon_I,1+epsilon_R,epsilon_I);
//wavevector, unit: 1/nm
double const kx_I=k_0*sqrtImag(epsilon_R,epsilon_I,1+epsilon_R,epsilon_I);

double const ky0_R=k_0*sqrtReal(-1,0,1+epsilon_R,epsilon_I);
//decay constant, kappa,in air
double const ky0_I=k_0*sqrtImag(-1,0,1+epsilon_R,epsilon_I);
//oscillation in air
double const ky2_R=k_0*sqrtReal(epsilon_I*epsilon_I-
    epsilon_R*epsilon_R,-2*epsilon_R*epsilon_I,1+epsilon_R,epsilon_I);
//decay in metal
double const ky2_I=k_0*sqrtImag(epsilon_I*epsilon_I-
    epsilon_R*epsilon_R,-2*epsilon_R*epsilon_I,1+epsilon_R,epsilon_I);
//oscillation in metal
double const lambda=2*PI/kx_R;//SPP wavelength along x(propagation direction)
//double mu_r=1;//dimensionless, relative permeability constant to mu_0;
//double impedance_0=376.730;//units: Ohm, sqrt(epsilon_0/mu_0)

/*-----FDTD parameters-----*/
int const Ntotal=100000;

```

```

//100000;//time steps number>3    41,141,241
int filestep=Ntotal/10;
//file written every "filestep" steps. Max=Ntotal, Min=1
int const halfgridsI=500;
//just for convenience, half incident area
int const gridsI=halfgridsI*2;
//simulation fields area
int const halfgridsJ=150;
//simulation fields area
int const gridsJ=halfgridsJ*2;
//simulation fields area—————>x (propagation direction)
int const CPML_n=30;
//Perfect Match Lay GRIDS width (30 GRIDS on each side)
int const TFSF_n=2;
//Total-Field/Scattered-Field technique layer width
int const SII=gridsI+CPML_n*2+TFSF_n*2;
//space GRIDS number, x axis
int const SJJ=gridsJ+CPML_n*2+TFSF_n*2;
//space GRIDS number, y axis
//int SKK=SII;//space GRIDS number, z axis (only for 3D, not in use)
//z (H-field direction for TM mode, dependency not in use for 2D case)
int const centerI=SII/2;
int const centerJ=SJJ/2;
int const metalJ=SJJ/2;//metal-air boundary for y
int const SFt=SJJ-CPML_n;//Top boundary for Scattered fields
int const SFb=CPML_n;//Bottom, Scattered Fields
int const SFl=CPML_n;//Left, Scattered fields
int const SFr=SII-CPML_n;//Right boundary for Scattered fields
int const TFt=SFt;//centerJ+halfgridsJ;//top, Total fields
int const TFb=SFb;//centerJ-halfgridsJ;//bottom, T fields
int const TFl=SFl+TFSF_n;//centerI-halfgridsI;//left
int const TFr=SFr;//centerI+halfgridsI;//right

```

```

int const Fileoffset=0*TFSF_n+0*CPML_n;// writing index
int const nonlinearM=100;//iteration loop number for nonlinear case 3;

double const ss_0=1/SQR2;
//stability constant for 2D case,dt half index shall be considered

//double const omega_0=2*PI*1000E-6;
//double const lambda_0=2*PI*c_0/omega_0;
short Nlambday=50;//grids sampling density,at least 20,
short Nlambdax=200;//related to dt calculation, at least 50
double dymax=lambda_0/Nlambday;
double dxmax=lambda_0/Nlambdax;
double maxVSmin=lambda_0*ky2_R;
//ratio of dmax:dmin, wavelength/penetration in metal,~10
int varygrade_n=40;//number of vary nonuniform grids for metal
int mingrade_n=50;//number of minimum grids
double dxmin=dxmax;//*pow(gradeRatex,gradeMinLeft-gradeLeft);
double dymin=dymax/maxVSmin;//*pow(gradeRatey,gradeMinDown-gradeDown);
double gradeRatex=1;
double gradeRatey=pow(maxVSmin,1.0/varygrade_n);
//make sure adjacent cells width ratio between 0.5~2

double const dt=1.0/(c_0*sqrt(1/dxmin/dxmin+1/dymin/dymin));
//unit: atto second,use the smaller cell value,maximum dt value
double const phi=0;
//unit: radian,angle of incidence wave vector to x-axis,
//-90~90 degree for this program
//*****FDTD extern parameters end here*****/

//for units, c:0.3nm/attosecond, dt:attosec, dx:nm, f:1MTHz
double const a_debye=0;//(2*tau_p-dt)/(2*tau_p+dt);
double const b_debye=0;//epsilon_0*depsilon_p*dt/(2*tau_p+dt);

```



```

double const a_lorentz=0;//(2-omega_p*omega_p*dt*dt)/(delta_p*dt+1);
double const b_lorentz=0;//(delta_p*dt-1)/(delta_p*dt+1);
double const c_lorentz=0;
//(epsilon_0*depsilon_p*omega_p*omega_p*dt*dt)/(delta_p*dt+1);
double const drudeFDTDdenom=2+dt*gamma_p;//denominator in P[n+1] calculation
double const a_drude=4/drudeFDTDdenom;
double const b_drude=(dt*gamma_p-2)/drudeFDTDdenom;
double const c_drude=epsilon_0*2*omega_p*omega_p*dt*dt/drudeFDTDdenom;
//epsilon_0 included!
double const a_raman=0;//(2-omega_raman*omega_raman*dt*dt)/(delta_raman*dt+1);
double const b_raman=0;//(delta_raman*dt-1)/(delta_raman+1);
double const c_raman=0;
//((1-alpha)*chi3*omega_raman*omega_raman*dt*dt)/(delta_raman*dt+1);

int SFflag(int ii,int jj);
//Boolean flag. True 1:ii,jj in scattered fields area. False 0:not
int TFflag(int ii,int jj);
//Boolean flag. True 1:ii,jj in total fields area. False 0:not
int CPMLflag(int ii,int jj);
//Boolean flag. True 1:ii,jj in CPML area. False 0:not
int Metalflag(int ii,int jj);
int Airflag(int ii,int jj);

/**MAIN***/
int main(){//return value=1:error. return value=0:normal exit.
    int nn=0;
    int ii=0;
    int jj=0;//ii,jj for space, nn for time
    int pp=0;//for pole loops
    int np0=0,np1=1,np2=2;//parity of nn,as time index flag
    double nT=2*PI/omega_0/dt;//period of plane wave, in terms of time steps.
    int FileFlag=0;

```

```

double ydis=0;//abs distance to origin point x=0,y=0
// double xdis=0;

double **Hzin=array2D(2,SJJ+1);//1. Incident wave source , ADE
double **Eyin=array2D(2,SJJ+1);//2. Incident wave source , ADE

double ***Hz=array3D(3,SII+1,SJJ+1);
//3. space cell[2][x-grid index][y-grid index] for Hz which takes the edges
double ***Ey=array3D(3,SII+1,SJJ+1);//4. space cell[2][x][y] for Ey,
double ***Ex=array3D(3,SII+1,SJJ+1);//5.
double ***Dy=array3D(3,SII+1,SJJ+1);//6.2 Displacement y
double ***Dx=array3D(3,SII+1,SJJ+1);//7.2 Displacement x
double ***Hzx=array3D(3,SII+1,SJJ+1);//8.Hz component
double ***Hzy=array3D(3,SII+1,SJJ+1);//9.Hz component

double ***Px_Debye=array3D(3,SII+1,SJJ+1);//10, polarization terms
double ***Px_Lorentz=array3D(3,SII+1,SJJ+1);//11
double ***Px_Drude=array3D(3,SII+1,SJJ+1);//12
double ***Py_Drude=array3D(3,SII+1,SJJ+1);//13
double ***Px_Kerr=array3D(3,SII+1,SJJ+1);//14
double ***Px_Raman=array3D(3,SII+1,SJJ+1);//15
double ***Py_Debye=array3D(3,SII+1,SJJ+1);//16
double ***Py_Lorentz=array3D(3,SII+1,SJJ+1);//17
double ***Py_Kerr=array3D(3,SII+1,SJJ+1);//18
double ***Py_Raman=array3D(3,SII+1,SJJ+1);//19
double ***Sp=array3D(3,SII+1,SJJ+1);//20

double **epsilon=array2D(SII+1,SJJ+1);
//21.dimensionless ,relative dielectric constant
double **mu=array2D(SII+1,SJJ+1);
//22.dimensionless , relative permeability
double *aax=array1D(SII+1);double *bbx=array1D(SII+1);

```

```

double *ccx=array1D(SII+1);//23 CPML coefficients
double *aay=array1D(SJJ+1);double *bby=array1D(SJJ+1);
double *ccy=array1D(SJJ+1);//24 CPML coefficients
// double *a_Hz=array1D(SII+1);double *b_Hz=array1D(SII+1);
// double *c_Hz=array1D(SII+1);// CPML coefficients
double *kkx=array1D(SII+1);double *kky=array1D(SJJ+1);
//double *k_Hz=array1D(SII+1);//25 CPML coefficients

double *sigmax=array1D(SII+1);//26.conductivity component, CPML terms
double *sigmay=array1D(SJJ+1);//27.conductivity component
// double **sigmax=array2D(SII+1,SJJ+1);
//27.1.magnetic loss component (fictional)
double ***psi_dyHz=array3D(3,SII+1,SJJ+1);//28.CPML
double ***psi_dxHz=array3D(3,SII+1,SJJ+1);//29.CPML
double ***psi_dxEy=array3D(3,SII+1,SJJ+1);//30.CPML
double ***psi_dyEx=array3D(3,SII+1,SJJ+1);//31.CPML

double *dx=array1D(SII+1);
//32.units: nm, grids size, 1/Nlambday of wavelength 500nm,
//typical optical value (500nm) ~ 25nm
double *dy=array1D(SJJ+1);//33
double *hxi=array1D(SII+1);//34 dual edge
double *hyj=array1D(SJJ+1);//35
double *xx=array1D(SII+1);//36 abs x-axis value
double *yy=array1D(SJJ+1);//37

double *HsourceAmp=array1D(SII+1);//38.output terms
double *EHphase=array1D(SII+1);//39. shift phase to match E
double **HzAmp=array2D(SII+1,SJJ+1);
//40. Average intensity of Hz, 2 periods average after being in steady state
double **ExAmp=array2D(SII+1,SJJ+1);//41.
double **EyAmp=array2D(SII+1,SJJ+1);//42.

```

```

double ***ET2=array3D(3,SII+1,SJJ+1);//43.4 |E|^2
double **DetectorsR4=array2D(6*(int)nT+2,SJJ+1);
//Detectors in real time one Rightside 4/4 of SII
double **DetectorsT=array2D(6*(int)nT+2,SII+1);//Detectors on Top
double **DetectorsB=array2D(6*(int)nT+2,SII+1);//Detectors on Bottom
double **DetectorsR1=array2D(6*(int)nT+2,SJJ+1);
//Detectors in real time one Rightside 1/4 of SII position
double **DetectorsR2=array2D(6*(int)nT+2,SJJ+1);
//Detectors in real time one Rightside 2/4 of SII position
double **DetectorsR3=array2D(6*(int)nT+2,SJJ+1);
//Detectors in real time one Rightside 3/4 of SII position
FullCubicFormula cubicRoot;//to solve for E from D analytically

FILE *fp_Hz; /* declare a File Pointer for Hz */
FILE *fp_Ey; /* declare a File Pointer for Ey */
FILE *fp_Ex;
FILE* fp_ADE;//1-D auxiliary grids;
FILE *fp_info;//additional info file
FILE* fp_Spectrum;

short file_importflag=1;//1 for import data to continue
short file_exportflag=0;//1 for ready to export all final fields data
short file_fullflag=0;//1 for additional files output

double Hup,Hdown,Hleft,Hright;//for real values: up,down,right,left
double Eup,Edown,Eleft,Eright;
//substitution for Hz and E's in Yee formulae,
//especially for TFSF_n calculation
int distance_n=0;
//distance in terms of cell number, from a point to bottom left corner of TF

```

```

/**Startline of main program body-----*/
/*****Startline of ALL cell value initialization-----*/
/*****Startline of nonuniform cell generator-----*/
if ((gradeRatey >=1.4)|| (gradeRatey <=0.6)){
    cout<<gradeRatey<<endl;

    cout<<
        "Varying cell step too big! Verify parameters for \
        non-uniform grids!\n\n";
    cout<<"Press 'q' to quit, or others to continue.\n\n";
    char gradequit;
    cin>>gradequit;
    if (gradequit=='q')
        exit(0);
    }

for(ii=0;ii<=SII;ii++){//dx,hxi initialization to uniform grid value
    dx[ii]=dxmax;//dx, dy are dividing Hz field
}

for(jj=0;jj<=SJJ;jj++){//dy,hyj initialization to uniform grid value
    dy[jj]=dymax;
}

/* if(0){//nonuniform grid mesh,x
    for(ii=gradeLeft;ii<gradeMinLeft;ii++){//grade down
        dx[ii]=dxmax*pow(gradeRatex,ii-gradeLeft);
    }
    for(ii=gradeMinLeft;ii<gradeMinRight;ii++){//min basin
        dx[ii]=dxmin;
    }
    for(ii=gradeMinRight;ii<gradeRight;ii++){//grade up
        dx[ii]=dxmax*pow(gradeRatex,gradeRight-ii);
    }
}

```

```

    }// if ,x
*/ int levelYcenter=centerJ;//basin valley center
int levely0=levelYcenter-mingrade_n-varygrade_n;//varying low limit
int levely1=levelYcenter-mingrade_n;//basin low edge
int levely2=levelYcenter+mingrade_n;//basin high edge
int levely3=levelYcenter+mingrade_n+varygrade_n;//varying high limit

if(1){//nonuniform grids,y
    for(jj=levely0;jj<levely1;jj++){//low ,dy decreasing
        dy[jj]=dymax*pow(gradeRatey ,levely0-jj);
//        cout<<jj<<"\t"<<dy [jj]<<endl;
    }
    for(jj=levely1;jj<levely2;jj++){//middle , ,dy const
        dy[jj]=dymin;
//        cout<<jj<<"\t"<<dy [jj]<<endl;
    }
    for(jj=levely2;jj<levely3;jj++){//high ,dy increasing
        dy[jj]=dymax*pow(gradeRatey ,jj-levely3);
//        cout<<jj<<"\t"<<dy [jj]<<endl;
    }
}

}

for(ii=1;ii<=SII;ii++){//dx ,hxi initialization to uniform grid value
    hxi [ii]=(dx [ii]+dx [ii-1])/2;//hxi ,hyj are dividing Ex ,Ey fields
    xx [ii]=xx [ii-1]+dx [ii];
}
hxi [0]=dx [0];
for(jj=1;jj<=SJJ;jj++){//dy ,hyj initialization to uniform grid value
    hyj [jj]=(dy [jj]+dy [jj-1])/2;
    yy [jj]=yy [jj-1]+dy [jj];
}
hyj [0]=dy [0];

```

```

double dxTF1=dx[TF1]; // for Hz correction
double hxiTF1_1=hxi[TF1-1]; // for Dy correction
/*--end of nonuniform cell generator******/

/*****start of CPML parameters-----*/
double sigma_max=0.0005;//0.08-0.0003
double kxx_max=1.50;//s=1/(k_w+sigma_w/(a_w+j*omega*epsilon_0)), w=x,y,z
double aax_max=0.001;//keep this term zero for most cases!
for(ii=0;ii<=SII;ii++){
    if(ii<=SF1){ //CPML left side
        double powerleft=pow((double)(SF1-ii)/CPML_n,2);
        kxx[ii]=1+(kxx_max-1)*powerleft;
        sigmax[ii]=powerleft*sigma_max;
        aax[ii]=aax_max*pow(((double)ii)/CPML_n,2);
        bbx[ii]=exp(-dt*(sigmax[ii]/epsilon_0/kxx[ii]+aax[ii]/epsilon_0));
        ccx[ii]=(bbx[ii]-1)/(kxx[ii]+kxx[ii]*kxx[ii]*aax[ii]/sigmax[ii]);
        if(sigmax[ii]==0)
            ccx[ii]=0;
    }
    else if(ii>=SFr){ //CPML right side
        double powerright=pow((double)(ii-SFr)/CPML_n,2);
        sigmax[ii]=powerright*sigma_max;
        kxx[ii]=1+(kxx_max-1)*powerright;
        aax[ii]=aax_max*pow((double)(SII-ii)/CPML_n,2);
        bbx[ii]=exp(-dt*(sigmax[ii]/epsilon_0/kxx[ii]+aax[ii]/epsilon_0));
        ccx[ii]=(bbx[ii]-1)/(kxx[ii]+kxx[ii]*kxx[ii]*aax[ii]/sigmax[ii]);
        if(sigmax[ii]==0)
            ccx[ii]=0;
    }
    else{//0
        sigmax[ii]=0;
        kxx[ii]=1;
    }
}

```

```

        aax [ ii ]=0;

    }
} // for ii

double kxTF1=kx [ TF1 ]; // for Hz, E correction

for ( jj=0; jj<=SJJ; jj++){

    if ( jj<=SFb ){ // CPML bottom

        double powerbottom=pow ( ( double ) ( SFb-jj ) / CPML_n , 2 );

        sigmay [ jj ]=powerbottom*sigma_max ;

        kky [ jj ]=1+(kx_max -1)*powerbottom ;

        aay [ jj ]=aax_max*pow ( ( ( double ) ( SFb-1)-jj ) / CPML_n , 2 );

        bby [ jj ]=exp ( -dt*(sigmay [ jj ] / epsilon_0 / kky [ jj ] + aay [ jj ] / epsilon_0 ) );

        ccy [ jj ]=(bby [ jj ] -1) / (kky [ jj ] + kky [ jj ] * kky [ jj ] * aay [ jj ] / sigmay [ jj ] );

        if ( sigmay [ jj ] ==0 )

            ccy [ jj ]=0;

    }

    else if ( jj>=SFt ){ // CPML top

        double powertop=pow ( ( double ) ( jj-SFt ) / CPML_n , 2 );

        sigmay [ jj ]=powertop*sigma_max ;

        kky [ jj ]=1+(kx_max -1)*powertop ;

        aay [ jj ]=aax_max*pow ( ( double ) ( SJJ-jj ) / CPML_n , 2 );

        bby [ jj ]=exp ( -dt*(sigmay [ jj ] / epsilon_0 / kky [ jj ] + aay [ jj ] / epsilon_0 ) );

        ccy [ jj ]=(bby [ jj ] -1) / (kky [ jj ] + kky [ jj ] * kky [ jj ] * aay [ jj ] / sigmay [ jj ] );

        if ( sigmay [ jj ] ==0 )

            ccy [ jj ]=0;

    }

    else {

        sigmay [ jj ]=0;

        kky [ jj ]=1;

        aay [ jj ]=0;

    }

} // for jj

```



```

double chi3[SII+1]={0};
for (ii=TF1;ii<=SII;ii++){
    chi3[ii]=chi3M;
    //(1-exp(-(ii-TF1)*(ii-TF1)/1.0/Nlambdax/Nlambdax))*chi3M;
    //chi3 ramp up
}
for (ii=0;ii<=SII;ii++){
    for(jj=0;jj<=SJJ;jj++){
        epsilon[ii][jj]=1.0;//medium initialization,relative values
        mu[ii][jj]=1.0;
    }//for(jj)
}//for(ii)
/*-----end of CPML parameters******/
/*-----Endline of ALL cell initialization******/

/*****Startline of FDTD major loops-----*/
double smoothupHc=0.001;
//unit: V/m artificial factor to tune up H field slowly
double smoothupEc=smoothupHc;
//together with chi(3) m2/v2, the intensity unit is W/m2
double smoothupH=0;//ini.
double smoothupE=0;//ini.
double decayfacy=1;//ini.along y-axis,decay term for SP
double phase_kx=atan2(kx_I,kx_R);
//phase difference due to wavevector Qx between H and E fields
double abs_kx=sqrt(kx_I*kx_I+kx_R*kx_R);
//absolute value of wavevector Qx
short rampup=4;//total time steps to ramp up the fields
double *chi3ii;//nonlinear susceptibility
double *Exnp1,*Hznp1,*Eynp1;//temp variables to speed up loop calculation
double *Exnp2,*Hznp2,*Eynp2;//temp variables to speed up loop calculation

```

```

double *Dxnp2,*Dynp2;// temp variables to speed up loop calculation
double *Px_Drudenp2,*Py_Drudenp2;
double *ET2np1,*ET2np2;
double *psi_dyHznp1,*psi_dxHznp1;
double *psi_dyHznp2,*psi_dxHznp2;
double *hyjjj,*hxiii,*dxii,*dyjj;
double *Px_Kerrnp1,*Py_Kerrnp1;
double *psi_dxEynp0,*psi_dyExnp0;
double yymetalJ=yy[metalJ];
//*****
for (nn=1;nn<=Ntotal;nn++){//time loop
    np0=(nn-1)%3;np1=nn%3;np2=(nn+1)%3;//parity conversion, 3-based for ADE

    smoothupH=smoothupHc*(1-exp(-(double)nn/rampup/nT));
    //for 1W laser on 2mmx2mm spot, field strength~1000 V/m
    smoothupE=smoothupEc*(1-exp(-(nn+0.5)/rampup/nT));
    //Boyd, Nonlinear Optics,Page3 E=5.14x10^11 V/m
    for (jj=0;jj<=metalJ;jj++){//SPP source, in Metal
        ydis=yy[jj]-yymetalJ;//ydis<0
        decayfacy=exp(ky2_R*ydis);
        //Hz and Ey have the same y coordinates position
        Hzin[nn%2][jj]=smoothupH*sin(omega_0*nn*dt)*decayfacy;
        //Hz, at step n, position i//
        Eyin[nn%2][jj]=smoothupE*sin(omega_0*(nn+0.5)*dt+kx_R*dxTF1*0.50-
            phase_kx)*decayfacy*abs_kx/omega_0/epsilon_0/epsilon_inf;
        //Dy, at step n+1/2, position i-1/2

    }

    for (jj=metalJ+1;jj<=SJJ;jj++){//SPP source, in Air
        ydis=yy[jj]-yymetalJ;//
        decayfacy=exp(-ky0_R*ydis);
        Hzin[nn%2][jj]=smoothupH*sin(omega_0*nn*dt)*decayfacy;
    }
}

```

```

//Hz, at step n, position i
Eyin [nn%2][jj]=smoothupE* sin (omega_0*(nn+0.5)*dt+kx_R*dxTF1*0.50 -
    phase_kx)*decayfacy*abs_kx/omega_0/epsilon_0;
//Dy, at step n+1/2, position i-1/2
}

for(ii=0;ii<=SII-1;ii++){//Terms w/o E[n+1]
    for(jj=0;jj<=SJJ-1;jj++){
        Hup=Hz[np2][ii][jj+1];//H, named relative to E
        Hdown=Hz[np2][ii][jj];
        Hright=Hz[np2][ii+1][jj];//H, named relative to E
        Hleft=Hz[np2][ii][jj];
        chi3ii=&chi3[ii];
        ET2np1=&ET2[np1][ii][jj];
        Exnp1=&Ex[np1][ii][jj];
        Eynp1=&Ey[np1][ii][jj];
        Px_Drudemp2=&Px_Drude[np2][ii][jj];
        Py_Drudemp2=&Py_Drude[np2][ii][jj];
        Px_Kerrnp1=&Px_Kerr[np1][ii][jj];
        Py_Kerrnp1=&Py_Kerr[np1][ii][jj];
        /*---Polarization calc starts-----*/
        if ((jj<=metalJ)){//in metal
            //-----terms include E[n]-----
            *Px_Drudemp2=a_drude*Px_Drude[np1][ii][jj]+b_drude*
                Px_Drude[np0][ii][jj]+c_drude*( *Exnp1);
            //c_drude includes Epsilon_0 already
            *Py_Drudemp2=a_drude*Py_Drude[np1][ii][jj]+b_drude*
                Py_Drude[np0][ii][jj]+c_drude*( *Eynp1);
//
            Px_Lorentz[np2][ii][jj]=0;
            //a_lorentz*Px_Lorentz[np1][ii][jj]+b_lorentz*
            //Px_Lorentz[np0][ii][jj]+c_lorentz*Ex[np1][ii][jj];
//
            Py_Lorentz[np2][ii][jj]=0;

```

```

//a_lorentz*Py_Lorentz [np1][ii][jj]+b_lorentz
//*Py_Lorentz [np0][ii][jj]+c_lorentz*Ey [np1][ii][jj];
//
Sp [np2][ii][jj]=0;//a_raman*Sp [np1][ii][jj]+b_raman*
//Sp [np0][ii][jj]+c_raman*(Ex [np1][ii][jj]*Ex [np1][ii][jj]+
// Ey [np1][ii][jj]*Ey [np1][ii][jj]);

//-----terms include E[n+1]-----
//
Px_Debye [np1][ii][jj]=0;
//a_debye*Px_Debye [np0][ii][jj]+b_debye*(Ex [np1][ii][jj]+
//Ex [np0][ii][jj]);
//
Py_Debye [np1][ii][jj]=0;
//a_debye*Py_Debye [np0][ii][jj]+b_debye*(Ey [np1][ii][jj]+
//Ey [np0][ii][jj]);
*Px_Kerrnp1=alpha*epsilon_0*(chi3ii)*(ET2np1)*(Exnp1);
//epsilon_0 included
*Py_Kerrnp1=alpha*epsilon_0*(chi3ii)*(ET2np1)*(Eynp1);
//
Px_Raman [np1][ii][jj]=epsilon_0*Ex [np1][ii][jj]*
//Sp [np1][ii][jj];
//
Py_Raman [np1][ii][jj]=epsilon_0*Ey [np1][ii][jj]*
//Sp [np1][ii][jj];
}//metal
else{//in air
//-----terms include E[n]
//
*Px_Drudenp2=0;
//
*Py_Drudenp2=0;
//
Px_Lorentz [np2][ii][jj]=0;
//
Py_Lorentz [np2][ii][jj]=0;
//
Sp [np2][ii][jj]=0;
//-----terms include E[n+1]
//
Px_Debye [np1][ii][jj]=0;
//
Py_Debye [np1][ii][jj]=0;
//
*Px_Kerrnp1=0;

```

```

//      *Py_Kerrnp1=0;
//      Px_Raman [np1][ii][jj]=0;
//      Py_Raman [np1][ii][jj]=0;
//      }//air
//
//
/*-----Polarization calc ends-----*/

/*-----Displacement field starts-----*/

for (ii=0;ii<=SII;ii++){//Dx loop-----
    for (jj=0;jj<=SJJ-1;jj++){
        Hup=Hz [np2][ii][jj+1];//H, named relative to E
        Hdown=Hz [np2][ii][jj];
        psi_dyHznp1=&psi_dyHz [np1][ii][jj];
        psi_dyHznp2=&psi_dyHz [np2][ii][jj];
        hyjjj=&hyj [jj];

        *psi_dyHznp2=bby [jj]*( *psi_dyHznp1)+ccy [jj]*( Hup-Hdown)/( *hyjjj);
        //CPML term, 0 unless in CPML layer
        Dx [np2][ii][jj]=Dx [np1][ii][jj]+dt*
            (( Hup-Hdown)/( *hyjjj)/kky [jj]+( *psi_dyHznp2));
    }//for (jj)
} //E loop(ii)

for (ii=0;ii<=SII-1;ii++){//Dy loop-----
    for (jj=0;jj<=SJJ;jj++){
        Hright=Hz [np2][ii+1][jj];//H, named relative to E
        Hleft=Hz [np2][ii][jj];
        psi_dxHznp2=&psi_dxHz [np2][ii][jj];
        psi_dxHznp1=&psi_dxHz [np1][ii][jj];
        hxiii=&hxi [ii];
    }
}

```

```

        *psi_dxHzn2=bbx [ ii ]*( *psi_dxHzn1 )+
        ccx [ ii ]*( Hright-Hleft )/( *hxiii );//CPML term
        Dy [ np2 ] [ ii ] [ jj ]=Dy [ np1 ] [ ii ] [ jj ]-dt*
        (( Hright-Hleft )/( *hxiii )/kxx [ ii ]+( *psi_dxHzn2 ));
    }//for ( jj )
} //E loop ( ii )

for ( jj=SFb; jj<=Sft; jj++){//Dy TFSF source correction, Hz:TF->SF
    Hright=Hz [ np2 ] [ TF1 ] [ jj ]-Hzin [ nn%2 ] [ jj ];//H, named relative to E
    Hleft=Hz [ np2 ] [ TF1 -1 ] [ jj ];
    Dy [ np2 ] [ TF1 -1 ] [ jj ]=Dy [ np1 ] [ TF1 -1 ] [ jj ]+dt*( Hleft-Hright )/hxiTF1_1;
//
    Dy [ np2 ] [ TF1 -1 ] [ jj ]=( exp(-tau*0.5*dt)*Dy [ np1 ] [ TF1 -1 ] [ jj ]+
    //dt*( Hleft-Hright )/hxi [ TF1 -1 ]/( exp(tau*0.5*dt) );
}

for ( ii=0; ii<=SII; ii++){
    //Ex Ey loop-----t=n+1/2-----
    for ( jj=0; jj<=SJJ; jj++){
        Exnp2=&Ex [ np2 ] [ ii ] [ jj ];//to be calc
        Eynp2=&Ey [ np2 ] [ ii ] [ jj ];//to be calc
        Dxn2=&Dx [ np2 ] [ ii ] [ jj ];//fix , from H
        Dynp2=&Dy [ np2 ] [ ii ] [ jj ];//fix , from H
        if ( Metalflag ( ii , jj )){
            ET2np2=&ET2 [ np2 ] [ ii ] [ jj ];
//
            ET2np1=&ET2 [ np1 ] [ ii ] [ jj ];
            Px_Drudemp2=&Px_Drude [ np2 ] [ ii ] [ jj ];
            Py_Drudemp2=&Py_Drude [ np2 ] [ ii ] [ jj ];
            chi3ii=&chi3 [ ii ];
            double cubic_a0 , cubic_a , cubic_b , cubic_c , AA;//AA=E*E
            cubic_a0=square ( *chi3ii );
            cubic_a=2*epsilon_inf *( *chi3ii );
            cubic_b=epsilon_inf *epsilon_inf ;

```

```

cubic_c=-((square(*Dxnp2-*Px_Drudenp2)+square(*Dynp2-
*Py_Drudenp2))/(epsilon_0*epsilon_0);
cubicRoot.setFormula(cubic_a0,cubic_a,cubic_b,cubic_c);
AA=real(cubicRoot.cubicRoot1());
*ET2np2=AA;
//
cout<<"a,b,c\t"<<cubic_a<<cubic_b<<cubic_c<<endl;
//
cout<<AA<<endl;
*Exnp2=((*Dxnp2)-(*Px_Drudenp2))/epsilon_0/
(epsilon_inf+(*chi3ii)*AA);
*Eynp2=((*Dynp2)-(*Py_Drudenp2))/epsilon_0/
(epsilon_inf+(*chi3ii)*AA);
//
*ET2np2=*ET2np1;
//
for(int mm=1;mm<=nonlinearM;mm++){
//Ex nonlinear iteration, for P(w=w+w-w) only
//
double Dynp2compare;
//
Dynp2compare=*Dynp2;
////
for(mm=0;fabs(Dynp2compare-*Dynp2)>=0.0000;mm++){
//
*Exnp2>(*Dxnp2/*-a_debye*Px_Debye[np1][ii][jj]-
//b_debye*Ex[np1][ii][jj]-Px_Lorentz[np2][ii][jj]*/-
//(*Px_Drudenp2))/(epsilon_0*epsilon_inf/*+b_debye*/+
//alpha*epsilon_0>(*chi3ii)(*ET2np2)/*+
//epsilon_0*Sp[np2][ii][jj]*/);
//
*Eynp2>(*Dynp2/*-a_debye*Py_Debye[np1][ii][jj]-
//b_debye*Ey[np1][ii][jj]-Py_Lorentz[np2][ii][jj]*/-
//(*Py_Drudenp2))/(epsilon_0*epsilon_inf/*+
//b_debye*/+alpha*epsilon_0>(*chi3ii)(*ET2np2)/*+
//epsilon_0*Sp[np2][ii][jj]*/);
//
*ET2np2>(*Exnp2)(*Exnp2)+(*Eynp2)(*Eynp2);
//
Dynp2compare=*Eynp2*(epsilon_0*epsilon_inf+
//alpha*epsilon_0>(*chi3ii)(*ET2np2))+
//*Py_Drudenp2;
//compared, back calc value

```

```

//          if ((nn>500)&&(jj==metalJ)&&(ii==TF1+10)){
//
//          cout<<"nn, ii, jj=="<<nn<<"\t"<<ii<<"\t"<<jj<<endl;
//
//          //          system("pause");
//
//          cout<<"Ex, Ey"<<*Exnp2<<"\t"<<*Eynp2<<endl;
//
//          cout<<"Dynp2compare=="<<Dynp2compare<<endl;
//
//          cout<<"*Dynp2=="<<*Dynp2<<endl;
//
//          cout<<"Differ=="<<Dynp2compare-*Dynp2<<"\tOR\t"
//
//          //<<(Dynp2compare/(*Dynp2)-1)*100<<"%"<<endl<<endl;
//
//          }// if nn
//
//          }// for (mm iteration)

//
//
//          }// if metalflag
//
//          else if(Airflag(ii, jj)){// in air
//
//          *Exnp2>(*Dxnp2)/epsilon_0;
//
//          *Eynp2>(*Dynp2)/epsilon_0;
//
//          }// else if Airflag
//
//          }// for (jj)
//
//          }//E loop(ii)

for (ii=1;ii<=SII-1;ii++){//Hz loop
    for (jj=1;jj<=SJJ-1;jj++){
        Eright=Ey[np2][ii][jj];
        Eleft=Ey[np2][ii-1][jj];
        Eup=Ex[np2][ii][jj];
        Edown=Ex[np2][ii][jj-1];
        psi_dxEynp0=&psi_dxEy[np0][ii][jj];
        psi_dyExnp0=&psi_dyEx[np0][ii][jj];
        dxii=&dx[ii];
        dyjj=&dy[jj];

        *psi_dxEynp0=bbx[ii]*psi_dxEy[np2][ii][jj]+

```



```

        ccx[ii]*(Erigh-Eleft)/( *dxii ); // CPML term
*psi_dyExnp0=bby[jj]*psi_dyEx[np2][ii][jj]+
        ccy[jj]*(Eup-Edown)/( *dyjj ); // CPML term
Hz[np0][ii][jj]=Hz[np2][ii][jj]-
        dt/mu_0*((1/kkx[ii]*(Erigh-Eleft)/( *dxii )-
        1/kky[jj]*(Eup-Edown)/( *dyjj )+*psi_dxEynp0-*psi_dyExnp0));
    } // for(jj)
} // Hz loop(ii)

if(1) { // Hz correction
    for(jj=SFb; jj<=Sft; jj++){ // Hzx TFsf correction, Ey:SF->TF
        Erigh=Ex[np2][TF1][jj];
        Eleft=Ex[np2][TF1-1][jj]+Eyin[nn%2][jj];
        Eup=Ex[np2][TF1][jj];
        Edown=Ex[np2][TF1][jj-1];

        Hz[np0][TF1][jj]=Hz[np2][TF1][jj]-
            dt/mu_0*((1/kkxTF1*(Erigh-Eleft)/dxTF1-
            1/kky[jj]*(Eup-Edown)/dy[jj]+psi_dxEy[np0][TF1][jj]-
            psi_dyEx[np0][TF1][jj]));
    }
}

double *ExAmpii,*EyAmpii,*HzAmpii;
if(nn>Ntotal-2*nT){
    // amplitude search, 2 period of time while being in steady state
    for(ii=0; ii<=SII; ii++){
        for(jj=0; jj<=SJJ; jj++){
            ExAmpii=&ExAmp[ii][jj];
            EyAmpii=&EyAmp[ii][jj];
            HzAmpii=&HzAmp[ii][jj];

            // HsourceAmp[jj]=max(HsourceAmp[jj], abs(Hzin[(nn+1)%2][jj]));

```

```

        *ExAmpii=max(fabs(Ex[np2][ii][jj]),*ExAmpii);
        *EyAmpii=max(fabs(Ey[np2][ii][jj]),*EyAmpii);
        *HzAmpii=max(fabs(Hz[np0][ii][jj]),*HzAmpii);

    }//jj
} //ii
} //amplitude search loop ends

if (nn>Ntotal-6*(int)nT){ //spectrum data detector
    for (jj=0;jj<=SJJ;jj++){
        DetectorsR4 [nn-Ntotal+6*(int)nT][jj]=Hz[np0][TFr-1][jj];
        DetectorsR1 [nn-Ntotal+6*(int)nT][jj]=Hz[np0][SII/4][jj];
        DetectorsR2 [nn-Ntotal+6*(int)nT][jj]=Hz[np0][SII/2][jj];
        DetectorsR3 [nn-Ntotal+6*(int)nT][jj]=Hz[np0][SII*3/4][jj];
    }
    for (ii=0;ii<=SII;ii++){
        DetectorsT [nn-Ntotal+6*(int)nT][ii]=Hz[np0][ii][SFt-1];
        DetectorsB [nn-Ntotal+6*(int)nT][ii]=Hz[np0][ii][SFb+1];
    }
}

/*-----Endline of FDTD major loops***** */

/***** Startline of .dat file writing-----*/
static clock_t lasttimeclock=clock();
if (nn%10==0) {
    cout<<"Time Step = "<<nn<<endl; //step screen display monitor
    if (nn%50==0){
        timer_yan(Ntotal,nn,50,lasttimeclock);
        lasttimeclock=clock();
    }
}
FileFlag=((nn%filestep==0)&&(nn!=0));

```

```

char namenumber [18]; // extension of output files

if (FileFlag) { // Hz auto file writing
    _itoa(nn, namenumber, 10);
    // convert time step into file name string, 10 based
    strcat(namenumber, ".dat");
    char file_Hz[18] = "Hz";
    strcat(file_Hz, namenumber); // define output (data) Hz
    fp_Hz = fopen(file_Hz, "w");
    // fprintf(fp_Hz, "x-axis\ty-axis\tHz\n");
    // fprintf(fp_Hz, "nm\tnm\tA.U.\n");

    // Hz, column output starts
    for (ii=Fileoffset; ii<=SII-Fileoffset; ii+=2){
        for (jj=Fileoffset; jj<=SJJ-Fileoffset; jj+=2){
            fprintf(fp_Hz, "%f\t%f\t%f\n", xx[ii], yy[jj], Hz[np2][ii][jj]);
            // column output format
        } // for jj
    } // for ii // Hz column output ends

    // Hz output, matrix output format starts
    /* for (jj=Fileoffset; jj<=SJJ-Fileoffset; jj++){ //
        for (ii=Fileoffset; ii<=SII-Fileoffset; ii++){
            {
                fprintf(fp_Hz, "%f\t", Hz[np2][ii][jj]);
            } // for ii
            fprintf(fp_Hz, "\n");
        } // for jj // matrix output end
    */
    printf("\t\t\tFile written:%s\n", file_Hz);

    // step indicator on screen

    fclose(fp_Hz);
} // file Hz writing ends (if)

```

```

if (FileFlag) { //Ey auto file writing
    itoa(nn,namenum,10);
    //convert time step into file name string, 10 based
    strcat(namenum, ".dat");
    char file_Ey[18]="Ey";
    strcat(file_Ey,namenum); //define output (data) Ey
    fp_Ey = fopen(file_Ey, "w");

//    fprintf(fp_Ey, "x-axis\ty-axis\tEy\n");
//    fprintf(fp_Ey, "nm\tnm\tA.U.\n");

    for (ii=Fileoffset;ii<=SII-Fileoffset;ii+=2){
        for (jj=Fileoffset;jj<=SJJ-Fileoffset;jj+=2){
            fprintf(fp_Ey,"%lf\t%lf\t%lf\n",xx[ii],yy[jj],Ey[np2][ii][jj]);
            //column output format
        }
    }

//for (jj=Fileoffset;jj<=SJJ-Fileoffset;jj++){ //Ey output
//    for (ii=Fileoffset;ii<=SII-Fileoffset;ii++)
//    {
//        fprintf(fp_Ey,"%f\t",Ey[np2][ii][jj]);
//    }
//    fprintf(fp_Ey,"\n");
//}

    fclose(fp_Ey);
} //file Ey writing ends (if)

if (FileFlag) { //Ex auto file writing
    itoa(nn,namenum,10);

```

```

//convert time step into file name string, 10 based
strcat(namenum,".dat");
char file_Ex[18]="Ex";
strcat(file_Ex,namenum);//define output (data) Ex
fp_Ex = fopen(file_Ex, "w");

//      fprintf(fp_Ex,"x-axis\ty-axis\tEx\n");
//      fprintf(fp_Ex,"nm\tnm\tA.U.\n");

for (ii=Fileoffset;ii<=SII-Fileoffset;ii+=2){
    for (jj=Fileoffset;jj<=SJJ-Fileoffset;jj+=2){
        fprintf(fp_Ex,"%lf\t%lf\t%lf\n",xx[ii],yy[jj],Ex[np2][ii][jj]);
        //column output format
    }
}

//for (jj=Fileoffset;jj<=SJJ-Fileoffset;jj++){//Ex output
//  for (ii=Fileoffset;ii<=SII-Fileoffset;ii++)
//  {
//      fprintf(fp_Ex,"%f\t",Ex[np2][ii][jj]);
//  }
//      fprintf(fp_Ex,"\n");
//}

fclose(fp_Ex);
} //file Ex writing ends (if)

if(0){ //output dx structure
    fp_Hz=fopen("dxminmax.dat","w");
    for(ii=0;ii<=SII-1;ii++){
        fprintf(fp_Hz,"%d\t%lf\n",ii,dx[ii]);
    }
}

```

```

        fclose(fp_Hz);
    }

    if (FileFlag&&0) { // ADE file writing
        itoa(nn,namenum,10);
        //convert time step into file name string, 10 based
        strcat(namenum, ".dat");
        char file_ADE[18]="ADE_Hz";
        strcat(file_ADE,namenum);
        fp_ADE=fopen(file_ADE,"w");
//        fprintf(fp_ADE,"x axis\tWave\n");
//        fprintf(fp_ADE,"(cell)\t(AU)\n");
        for (ii=Fileoffset;ii<=SII-Fileoffset;ii++){
            fprintf(fp_ADE,"%d\t%lf\n",ii,Hzin[nn%2][ii]);
        }
        fclose(fp_ADE);
    } //file ADE writing ends (if)

} //for(nn)***Time loop ends*****/

printf("\nFinishing Program...\n\n");
if(1){ //Hz Amplitude output
    fp_Hz = fopen("HzAmp.dat", "w");
//    fprintf(fp_Hz,"x-axis\ty-axis\tHzAmp\n");
//    fprintf(fp_Hz,"nm\tnm\tA.U.\n");
    for (ii=Fileoffset;ii<=SII-Fileoffset;ii+=2){
        for (jj=Fileoffset;jj<=SJJ-Fileoffset;jj+=2){
            fprintf(fp_Hz,"%lf\t%lf\t%lf\n",xx[ii],yy[jj],HzAmp[ii][jj]);
            //column output format
        }
    }
}
}

```

```

/*      for (jj=Fileoffset ;jj<=SJJ-Fileoffset ;jj++){
          for (ii=Fileoffset ;ii<=SII-Fileoffset ;ii++)
            {
              fprintf(fp_Hz,"%f\t",HzAmp[ii][jj]);
//      Hzshift[ii][jj]=Hz[np0][ii][jj];
          }//matrix output format
          fprintf(fp_Hz,"\n");
        }
*/      fclose(fp_Hz);
    }

if(1){//Hz Time profile (for Fourier DetectorsR4 Analysis) output, Right Side
    fp_Spectrum=fopen("DetectorsR_Full.dat","w");
    fprintf(fp_Spectrum,"Time");
    for (jj=TFb+1;jj<=metalJ;jj+=2){
        fprintf(fp_Spectrum,"\t");
        fprintf(fp_Spectrum,"Metal"); //column output format
    }
    for (jj=metalJ+1;jj<=TFt-1;jj+=2){
        fprintf(fp_Spectrum,"\t");
        fprintf(fp_Spectrum,"Air"); //column output format
    }
    fprintf(fp_Spectrum,"\n");
    for(int iin=1;iin<=6*(int)nT;iin++){//dont change iin++
        fprintf(fp_Spectrum,"%lf",iin*dt/2/PI);
// 1/(2 PI) so that the units for FFT is rad/attosec
//(omega angular frequency rather than frequency)
        for (jj=TFb+1;jj<=TFt-1;jj+=2){
            fprintf(fp_Spectrum,"\t");
            fprintf(fp_Spectrum,"%lf",DetectorsR4[iin][jj]);

```

```

        //column output format
    }

    fprintf(fp_Spectrum, "\n");
}

fclose(fp_Spectrum);
}

if(1){ //Hz Time profile (for Fourier DetectorsR4 Analysis) output, Right Side
    fp_Spectrum=fopen("DetectorsR_Quarter.dat", "w");
    fprintf(fp_Spectrum, "Time");
    for (jj=TFb+1; jj<=metalJ; jj+=2){
        fprintf(fp_Spectrum, "\t");
        fprintf(fp_Spectrum, "Metal"); //column output format
    }
    for (jj=metalJ+1; jj<=TFt-1; jj+=2){
        fprintf(fp_Spectrum, "\t");
        fprintf(fp_Spectrum, "Air"); //column output format
    }
    fprintf(fp_Spectrum, "\n");
    for(int iin=1; iin<=6*(int)nT; iin++){
        fprintf(fp_Spectrum, "%lf", iin*dt/2/PI);
        // 1/(2 PI) so that the units for FFT is rad/attosec
        //(omega angular frequency rather than frequency)
        for (jj=TFb+1; jj<=TFt-1; jj+=2){
            fprintf(fp_Spectrum, "\t");
            fprintf(fp_Spectrum, "%lf", DetectorsR1[iin][jj]);
            //column output format
        }
        fprintf(fp_Spectrum, "\n");
    }
    fclose(fp_Spectrum);
}
}

```



```

if(1){//Hz Time profile (for Fourier DetectorsR1 Analysis) output, Right Side
    fp_Spectrum=fopen("DetectorsR_Half.dat","w");
    fprintf(fp_Spectrum,"Time");
    for (jj=TFb+1;jj<=metalJ;jj+=2){
        fprintf(fp_Spectrum,"\t");
        fprintf(fp_Spectrum,"Metal"); //column output format
    }
    for (jj=metalJ+1;jj<=TFt-1;jj+=2){
        fprintf(fp_Spectrum,"\t");
        fprintf(fp_Spectrum,"Air"); //column output format
    }
    fprintf(fp_Hz,"\n");
    for(int iin=1;iin<=6*(int)nT;iin++){
        fprintf(fp_Spectrum,"%lf",iin*dt/2/PI);
        // 1/(2 PI) so that the units for FFT is rad/attosec
        //(omega angular frequency rather than frequency)
        for (jj=TFb+1;jj<=TFt-1;jj+=2){
            fprintf(fp_Spectrum,"\t");
            fprintf(fp_Spectrum,"%lf",DetectorsR2[iin][jj]);
            //column output format
        }
        fprintf(fp_Spectrum,"\n");
    }
    fclose(fp_Spectrum);
}

if(1){//Hz Time profile (for Fourier DetectorsR2 Analysis) output, Right Side
    fp_Spectrum=fopen("DetectorsR_ThreeQuarters.dat","w");
    for (jj=TFb+1;jj<=metalJ;jj+=2){
        fprintf(fp_Spectrum,"Metal\t"); //column output format
    }
    for (jj=metalJ+1;jj<=TFt-1;jj+=2){
        fprintf(fp_Spectrum,"Air\t"); //column output format
    }
}

```

```

    }
    fprintf(fp_Hz, "\n");
    for(int iin=1; iin<=6*(int)nT; iin++){
        fprintf(fp_Spectrum, "%lf", iin*dt/2/PI);
        // 1/(2 PI) so that the units for FFT is rad/attosec
        //(omega angular frequency rather than frequency)
        for (jj=TFb+1; jj<=TFt-1; jj+=2){
            fprintf(fp_Spectrum, "\t");
            fprintf(fp_Spectrum, "%lf", DetectorsR3[iin][jj]);
            //column output format
        }
        fprintf(fp_Spectrum, "\n");
    }
    fclose(fp_Spectrum);
}

if(1){ //Hz Time profile (for Fourier DetectorsT Analysis) output, Top Edge
    fp_Spectrum=fopen("DetectorsT.dat", "w");

    for(int iin=1; iin<=6*(int)nT; iin++){
        fprintf(fp_Spectrum, "%lf", iin*dt/2/PI);
        // 1/(2 PI) so that the units for FFT is rad/attosec
        //(omega angular frequency rather than frequency)
        for (ii=centerI; ii<=SFr; ii+=2){
            fprintf(fp_Spectrum, "\t");
            fprintf(fp_Spectrum, "%lf", DetectorsT[iin][ii]);
            //column output format
        }
        fprintf(fp_Spectrum, "\n");
    }
    fclose(fp_Spectrum);
}
}

```

```

if(1){//Hz Time profile (for Fourier DetectorsB Analysis) output, Bottom Edge
    fp_Spectrum=fopen("DetectorsB.dat","w");

    for(int iin=1;iin<=6*(int)nT;iin++){
        fprintf(fp_Spectrum,"%lf",iin*dt/2/PI);
        // 1/(2 PI) so that the units for FFT is rad/attosec
        //(omega angular frequency rather than frequency)
        for (ii=centerI;ii<=SFr;ii+=2){
            fprintf(fp_Spectrum,"\t");
            fprintf(fp_Spectrum,"%lf",DetectorsB[iin][ii]);
            //column output format
        }
        fprintf(fp_Spectrum,"\n");
    }
fclose(fp_Spectrum);
}

if(1){//Ey Amplitude output
    fp_Ey = fopen("EyAmp.dat", "w");
//    fprintf(fp_Hz,"x-axis\ty-axis\tEyAmp\n");
//    fprintf(fp_Hz,"nm\tnm\tA.U.\n");
    for (ii=Fileoffset;ii<=SII-Fileoffset;ii+=2){
        for (jj=Fileoffset;jj<=SJJ-Fileoffset;jj+=2){
            fprintf(fp_Ey,"%lf\t%lf\t%lf\n",xx[ii],yy[jj],EyAmp[ii][jj]);
            //column output format
        }
    }
/*    for (jj=Fileoffset;jj<=SJJ-Fileoffset;jj++){//Ey Intensity output
        for (ii=Fileoffset;ii<=SII-Fileoffset;ii++)
            {

```

```

        fprintf(fp_Ey,"%f\t",EyAmp[ ii ][ jj]);
    }

    fprintf(fp_Ey,"\n");

}

*/    fclose(fp_Ey);
}

if(1){//Ex Amplitude output
    fp_Ex = fopen("ExAmp.dat", "w");
//    fprintf(fp_Ex,"x-axis\ty-axis\tEx\n");
//    fprintf(fp_Ex,"nm\tnm\tA.U.\n");
    for (ii=Fileoffset;ii<=SII-Fileoffset;ii+=2){
        for (jj=Fileoffset;jj<=SJJ-Fileoffset;jj+=2){
            fprintf(fp_Ex,"%lf\t%lf\t%lf\n",xx[ ii ],yy[ jj ],ExAmp[ ii ][ jj]);
            //column output format
        }
    }
/*    for (jj=0+Fileoffset;jj<=SJJ-Fileoffset;jj++){//Ex Intensity output
        for (ii=0+Fileoffset;ii<=SII-Fileoffset;ii++)
        {
            fprintf(fp_Ex,"%f\t",ExAmp[ ii ][ jj]);
        }

        fprintf(fp_Ex,"\n");
    }
*/    fclose(fp_Ex);
}

if(0){//PML error test
    double error1=0;
    double error2=0;
    for(ii=SF1+1;ii<=SFr-1;ii++){
        error1+=(HzAmp[ ii ][ centerJ]-1)*(HzAmp[ ii ][ centerJ]-1);

```

```

        error2=
            error2>abs(HzAmp[ii][centerJ]-1)?error2:abs(HzAmp[ii][centerJ]-1);
    }
    cout<<"Total error=\t"<<sqrt(error1)<<endl;
    cout<<"Max error=\t"<<error2<<endl;
}

if(0){//ADE source Amplitude output
    fp_Ex = fopen("HsourceAmp.dat", "w");
    for(ii=SFl+1;ii<=SFr-1;ii++){
        fprintf(fp_Ex,"%d\t%f",ii,(HzAmp[ii][centerJ]-1)*100);
        fprintf(fp_Ex,"\n");
    }
    fclose(fp_Ex);
}

if(0){//ADE source wave output
    fp_Ex = fopen("Hzin.dat", "w");
    for (jj=CPML_n+Fileoffset;jj<=SJJ-CPML_n-Fileoffset;jj++)
    {
        fprintf(fp_Ex,"%d\t%f",yy[jj],Hzin[nn%2][jj]);
//        EHphase[ii]=(Hzin[(nn+1)%2][ii]);//+Hzin[nn%2][ii]/2.0;
        //time average
        fprintf(fp_Ex,"\n");
    }
    fclose(fp_Ex);
}

if(0){//phase info, test
    fp_Hz = fopen("EHphase.dat", "w");
    fprintf(fp_Hz,"ii\tEy\tEyPhase\tHz\tHzPhase\n");
    fprintf(fp_Hz,"time\tA.U.\tDegree\tA.U.\tDegree\n");
}

```

```

    for (ii=Fileoffset;ii<=SII-Fileoffset;ii++)
    {

        }//matrix output format

        fprintf(fp_Hz," \n");

    fclose(fp_Hz);

}

if(1){//program info output

    fp_info=fopen(" info.txt","w");

    fprintf(fp_info," \n-----Physics \
                parameters-----\n\n");

    fprintf(fp_info," epsilon_0=%f (nF/m)\n",epsilon_0);
    fprintf(fp_info," mu_0=%f (nH/m)\n",mu_0);
    fprintf(fp_info," c_0=%f (nm/attosec)\n",c_0);
    fprintf(fp_info," lambda_0=%f (nm)\n",lambda_0);
    fprintf(fp_info," omega_0=%f (rad/attosec)\n",omega_0);
    fprintf(fp_info," Frequency f_0=%e(GGHz)\n",omega_0/2./PI);
    fprintf(fp_info," k_0=%f (/nm)\n",k_0);
    fprintf(fp_info," epsilon_inf=%f\n\n",epsilon_inf);
    fprintf(fp_info," (Debye)\tdepsilon_p=%f\n",depsilon_p);
    fprintf(fp_info," (Debye)\ttau_p=%f\n",tau_p);
    fprintf(fp_info," (Lorentz)\tdelta_p=%f\n",delta_p);
    fprintf(fp_info," (Drude)\tomega_p=%f\n",omega_p);
    fprintf(fp_info," (Drude)\tgamma_p=%f\n",gamma_p);
    fprintf(fp_info," (Kerr)\talpha=%f\n",alpha);
    fprintf(fp_info," (Kerr)\tchi(3)=%e (m2/V2)\n",chi3M);
    fprintf(fp_info," (Raman)\ttau1=%f\n",tau1);
    fprintf(fp_info," (Raman)\ttau2=%f\n",tau2);
    fprintf(fp_info," (Raman)\tomega_raman=%f\n",omega_raman);
    fprintf(fp_info," (Raman)\tdelta_raman=%f\n\n",delta_raman);
    fprintf(fp_info," For metal, Re(epsilon)=%f\tIm(epsilon)=%f\

```

```

        \n",epsilon_R,epsilon_I);
fprintf(fp_info," Hz input amplitude:\t%E (A/m)\n",smoothupHc);
fprintf(fp_info," SPP wave length in medium lambda=%f(nm)\n",lambda);
fprintf(fp_info," SPP wave vector Re(kx)=%f(1/nm)\tIm(kx)=%e(1/nm)\n\
        \n",kx_R,kx_I);
fprintf(fp_info," Evanescece wave in air\t Re(ky0)=%f(1/nm)\tIm(ky0)=%e\
        (1/nm)\n Or penetration depth\t 1/Re(ky0)=%f(nm)\t1/Im\
        (ky0)=%f(nm)\n",ky0_R,ky0_I,1./ky0_R,1./ky0_I);
fprintf(fp_info," \nEvanescece wave in metal\t \
        Re(ky2)=%f(1/nm)\tIm(ky2)=%e(1/nm)\n \
        Or penetration depth\t 1/Re(ky2)=%f(nm)\t1/Im(ky2)=\
        %f(nm)\n",ky2_R,ky2_I,1./ky2_R,1./ky2_I);

fprintf(fp_info," Ey input amplitude (V/m):\t(in air) %E\
        \t(in metal) %E\n",\
        smoothupEc*abs_kx/omega_0/epsilon_0/epsilon_inf,\
        smoothupEc*abs_kx/omega_0/epsilon_0);
fprintf(fp_info," Incident wave period:%d time steps(=%f fs)\n\n",
        int(nT),nT*dt/1000);

fprintf(fp_info," \n\n\n-----FDTD \
        parameters-----\n\n");

fprintf(fp_info,
"Total time simulated: steps=%d(=%f fs)\n",Ntotal,dt*Ntotal/1000.);
fprintf(fp_info," dt=%f attosecond\n",dt);
fprintf(fp_info," File written every %d time steps(=%f fs)\
        \twith (ABC frame) offset=%d\n\n\n",
        filestep,dt*filestep/1000.,Fileoffset);

fprintf(fp_info," Total cell numbers along x: %d\tty: %d\n",SII,SJJ);
fprintf(fp_info," Grids (coarse) sampling density:(x)%d\t(y)%d\n",

```

```

Nlambdax , Nlambday );
fprintf(fp_info , " dxmax=%f nm\tdxmin=%f nm\tx-(distance)total=%f nm\n" ,
        dxmax , dxmin , xx[SII] );
fprintf(fp_info , " dymax=%f nm\tdymin=%f nm\ty-(distance)total=%f nm\n" ,
        dymax , dymin , yy[SJJ] );
fprintf(fp_info , " Air-Metal interface at y(J)=%d(+) ~ %f(nm)\n" ,
        metalJ , yy[metalJ] );
fprintf(fp_info , " Total field frame:\t (Left,Right,Bottom,Top)=\
        (%d,%d,%d,%d)( cell)~(%f,%f,%f,%f)(nm)\n" ,
        TF1 , TFr , TFb , TFt , xx[TF1] , xx[TFr] , yy[TFb] , yy[TFt] );
fprintf(fp_info , " Scattered field frame:\t \
        (Left,Right,Bottom,Top)=\
        (%d,%d,%d,%d)( cell)~(%f,%f,%f,%f)(nm)\n" ,
        SF1 , SFr , SFb , SFt , xx[SF1] , xx[SFr] , yy[SFb] , yy[SFt] );
fprintf(fp_info , " CPML layer width CPML_n=%d\n" , CPML_n);
fprintf(fp_info , "\nTotal nonuniform size numbers:\
        x %d\t y %d\n" , 0 , varygrade_n);
fprintf(fp_info , " size ratio of max to min cell maxVSmin=%f\n" , maxVSmin);
fprintf(fp_info , " size ratio of adjacent cells: %f or %f\n" ,
        gradeRatey , 1.0/gradeRatey);
fprintf(fp_info , " nonuniform area (y): %d\%\d---%d/%d\
        (cell)\t~\t%f\%\f---%f/%f (nm)\n" ,
        levely0 , levely1 , levely2 , levely3 , yy[levely0] ,
        yy[levely1] , yy[levely2] , yy[levely3] );
fprintf(fp_info , " Field Ramp Up %d ~wavelength(s)\n\n" ,
        (int)sqrt((double)rampup));

fprintf(fp_info , "\nndebye model FDTD parameters: a=%f\tb=%f\
        \n" , a_debye , b_debye);
fprintf(fp_info , " lorentz model FDTD parameters: a=%f\tb=%f\tc=%f\
        \n" , a_lorentz , b_lorentz , c_lorentz);
fprintf(fp_info , " drude model FDTD parameters: a=%f\tb=%f\tc=%f\

```



```

        \n" , a_drude , b_drude , c_drude );
fprintf(fp_info , "Raman model FDTD parameters : a=%f\tb=%f\tc=%f\n\
        \n" , a_raman , b_raman , c_raman );

fprintf(fp_info , "CPML factors : sigma_max=%f\tkx_max=%f\taax_max=\
        %f\n\n" , sigma_max , kx_max , aax_max );
fprintf(fp_info , "Nonlinear iteration loop number : nonlinearM=\
        %d\n\n" , nonlinearM);

fprintf(fp_info , "\n\n\n-----CPML \
        parameters-----\n\n");
fprintf(fp_info , " sigma_max=%f\n" , sigma_max);
fprintf(fp_info , " kx_max=%f\n" , kx_max);
fprintf(fp_info , " aax_max=%f\n" , aax_max);

fprintf(fp_info , "\n\n\n-----Time/Date \
        info-----\n\n");

time_t rawtime;
struct tm * timeinfo;
time ( &rawtime );
timeinfo = localtime ( &rawtime );
fprintf ( fp_info , "\nThis file was written at (date/time) %s" ,
        asctime ( timeinfo ) );

fclose(fp_info);
}

printf("\t\t\tFile written for %s\n" , "Average Intensity");
//step indicator on screen
/*-----Endline of .dat file writing******/

```

```

/*-----clean up-----*/
free3D(Hz,3,SII+1);//1
free3D(Hzx,3,SII+1);//2
free3D(Hzy,3,SII+1);//3
free3D(Dy,3,SII+1);//2.2 Displacement y
free3D(Dx,3,SII+1);//3.2 Displacement x
free3D(ET2,3,SII+1);//3.4 |E|^2
free3D(Ey,3,SII+1);//4
free3D(Ex,3,SII+1);//5
free3D(Px_Debye,3,SII+1);
free3D(Px_Lorentz,3,SII+1);
free3D(Px_Raman,3,SII+1);
free3D(Px_Kerr,3,SII+1);
free3D(Px_Drude,3,SII+1);
free3D(Py_Drude,3,SII+1);
free3D(Py_Debye,3,SII+1);
free3D(Py_Lorentz,3,SII+1);
free3D(Py_Kerr,3,SII+1);
free3D(Py_Raman,3,SII+1);
free3D(Sp,3,SII+1);
// cout<<"11"<<endl;
free2D(epsilon,SII+1);//6
free1D(sigmamax);//8
free1D(sigmay);//9
free2D(mu,SII+1);//7
// free2D(sigmastarx,SII+1);//10
free3D(psi_dyHz,3,SII+1);//11
free3D(psi_dxHz,3,SII+1);//
free3D(psi_dxEy,3,SII+1);//
free3D(psi_dyEx,3,SII+1);//
free1D(aax);free1D(bbx);free1D(ccx);
free1D(aay);free1D(bby);free1D(ccy);

```

```

// free1D(a_Hz);free1D(b_Hz);free1D(c_Hz);
free1D(kkx);free1D(kky);//free1D(k_Hz);
free2D(Hzin,2);//18
free2D(Eyin,2);//19
free2D(HzAmp,SII+1);//20
free2D(ExAmp,SII+1);//21
free2D(EyAmp,SII+1);//22
free2D(DetectorsR4,6*(int)nT+2);
free2D(DetectorsR1,6*(int)nT+2);
free2D(DetectorsR2,6*(int)nT+2);
free2D(DetectorsR3,6*(int)nT+2);
free2D(DetectorsT,6*(int)nT+2);
free2D(DetectorsB,6*(int)nT+2);
free1D(HsourceAmp);//23
free1D(EHphase);//24
//
free1D(dx);//38
free1D(dy);//39
free1D(hxi);//40
free1D(hyj);//41
free1D(xx);
free1D(yy);

printf("\n");
printf("File written successfully. Now closing it...\n");
printf("\n");

// system("PAUSE");
return EXIT_SUCCESS;//(main)
} //end main()
/****Endline of main program body*****/

```

```

double Ex_pml_func(double Ex_previous, double Hzup, double Hzdown, double sigma,
double epsilon, double d_x){//general E(n+1)calc in CPML_n layer
double CE=(epsilon-(sigma*dt*0.5/epsilon_0))/(epsilon+
(sigma*dt*0.5/epsilon_0));//factor for E
double CH=(dt/epsilon_0)/(epsilon+sigma*dt*0.5/epsilon_0)/d_x;//factor for H
double Ex=CE*Ex_previous+CH*(Hzup-Hzdown);
//H_2 has larger space index than H_1
return Ex;
}//Ex_pml_func

double Ey_pml_func(double Ey_previous, double Hzright, double Hzleft,
double sigma, double epsilon, double d_y){//general E(n+1)calc in CPML_n layer
double CE=(epsilon-(sigma*dt*0.5/epsilon_0))/(epsilon+
(sigma*dt*0.5/epsilon_0));
double CH=-(dt/epsilon_0)/(epsilon+sigma*dt*0.5/epsilon_0)/d_y;
double Ey=CE*Ey_previous+CH*(Hzright-Hzleft);
return Ey;
}//Ey_pml_func

double Hxz_func(double H_previous, double Eyright, double Eyleft, double sigmastar,
double mu, double d_x){//gneral H(n+1) calc in CPML_n layer
double CH=(mu-sigmastar*dt*0.5/mu_0)/(1+sigmastar*dt*0.5/mu_0);
double CE=-(dt/mu_0)/(mu+sigmastar*dt*0.5/mu_0)/d_x;
double Hz=CH*H_previous+CE*(Eyright-Eyleft);
return Hz;
}//Hxz_func

double Hzy_func(double H_previous, double Exup, double Exdown, double sigmastar,
double mu, double d_y){//gneral H(n+1) calc in CPML_n layer
double CH=(mu-sigmastar*dt*0.5/mu_0)/(mu+sigmastar*dt*0.5/mu_0);
double CE=(dt/mu_0)/(mu+sigmastar*dt*0.5/mu_0)/d_y;

```

```

    double Hz=CH*H_previous+CE*(Exup-Exdown);

    return Hz;
} //Hzy_func

void Hz_contour_func(double*** Hz,double*** Ex,double*** Ey,int time,int ii,
int jj,double area,double Fn,double Fs,double Fw,double Fe){

    //North,South,West,East;Integral Form of Maxwell Equations,contour calc
    int pretime=(time+2)%3;

    double En=Ex[pretime][ii][jj];
    double Es=Ex[pretime][ii][jj-1];
    double Ew=Ey[pretime][ii-1][jj];
    double Ee=Ey[pretime][ii][jj];

    Hz[time][ii][jj]=Hz[pretime][ii][jj]+dt/mu_0/area*((En*Fn-Es*Fs)*dxmax+
        (Ew*Fw-Ee*Fe)*dymax);
} //Hz_contour_func

int SFflag(int ii,int jj){

    int flag=(ii<=SFr)&&(ii>=SF1)&&(jj<=SFt)&&(jj>=SFb);

    //Boolean flag,1 for in Scattered fields area, 0 for NOT.

    return flag;
}

int TFflag(int ii,int jj){

    int flag=(ii<=TFr)&&(ii>=TF1)&&(jj<=TFt)&&(jj>=TFb);

    //Boolean flag,1 for in Total fields area, 0 for NOT.

    return flag;
}

int CPMLflag(int ii,int jj){

    int flag=(ii<SF1)&&(ii>SFr)&&(jj<SFb)&&(jj>SFt);

    //Boolean flag,1 for in CPML area, 0 for NOT.

    return flag;
}

```

```

}

int Metalflag(int ii,int jj){//determine if (ii,jj) is in metal
    int flag=(jj<=metalJ);//(ii<=SFr)&&(ii>=SF1)&&(jj<=metalJ)&&(jj>=SFb);
    return flag;
}

int Airflag(int ii,int jj){//determine if (ii,jj) is in air
    int flag=(jj>metalJ);//(ii<=SFr)&&(ii>=SF1)&&(jj>metalJ)&&(jj<=SFt);
    return flag;
}

```

FDTDarray.h

```

#ifndef FDTDarray_h
#define FDTDarray_h

double* array1D(int n);
double** array2D(int narrows ,int ncolumns);
double*** array3D(int narrows ,int ncolumns ,int nz);

void free1D(double* p1);
void free2D(double** p2,int nx);
void free3D(double*** p3,int nx, int ny);

double array_max(double* p,int from,int to);
double array_min(double* p,int from,int to);
double array_amp(double* p,int from,int to);

```

```
double Dx_func(double Dx_previous, double Hzxdown, double Hzxup, double hyj, double dt);
double Dy_func(double Dy_previous, double Hzyleft, double Hzyright,
double hxi, double dt);

double sqrtReal(double a, double b, double c, double d);
double sqrtImag(double a, double b, double c, double d);
double sqrtReal(double a, double b);
double sqrtImag(double a, double b);
double square(double a);
double cube(double a);

#endif
```

FDTDarray.cpp

```
/* Malloc_multi_array */
/* malloc to defined a 2 or 3 dimensional array*/
#include <stdio.h>
#include <iostream>
#include <stdlib.h>
#include <cmath>
#include "FDTDarray.h"
using namespace std;
/*-----end #include-----*/

/*-----start array1D function definition-----*/
double* array1D(int n)
{
    int i=0;
    double* p1;
    p1=(double*)malloc(n*sizeof(double));
    if(p1==NULL){
        fprintf(stderr,"out of memory\n");
        return 0;
    }// if

    for(i=0;i<=n-1;i++){
        p1[i]=0;
    }// for(i)

    return(p1);
} //array1D()
```



```

/*-----end array1D function definition-----*/

/*-----Start array2D function definition-----*/
double** array2D(int nrows,int ncolumns)
{
    int i=0;
    int j=0;
    double **p2;
    p2 = (double**)malloc(nrows * sizeof(double *));
    if(p2 == NULL){
        fprintf(stderr, "out of memory\n");
        return 0;
    }//(if)

    for(i = 0; i <= nrows-1; i++){
        p2[i] = (double*)malloc(ncolumns * sizeof(double));
        if(p2[i] == NULL){
            fprintf(stderr, "out of memory\n");
            return 0;
        }//(if)
    }// (for(i))

    for (i=0;i<=nrows-1;i++){
        for (j=0;j<=ncolumns-1;j++){
            p2[i][j]=0;
        }//(for(i))
    }//(for(j))
    return (p2);
} //array2D
/*-----end array2D definitino-----*/

```

```

/*-----Start array3D function definition-----*/
double*** array3D(int nrows,int ncolumns,int nz)
{
    int i=0;
    int j=0;
    int k=0;
    double ***p3;
    p3 = (double***)malloc(nrows * sizeof(double **));
    if(p3 == NULL){
        fprintf(stderr, "out of memory\n");
        return 0;
    }//(if)

    for(i = 0; i <= nrows -1; i++){
        p3[i] = (double**)malloc(ncolumns * sizeof(double*));
        if(p3[i] == NULL){
            fprintf(stderr, "out of memory\n");
            return 0;
        }//(if)
    }

    for(i = 0; i <= nrows -1; i++)
        for(int j=0;j<=ncolumns -1;j++){
            p3[i][j] = (double*)malloc(nz * sizeof(double));
            if(p3[i][j] == NULL){
                fprintf(stderr, "out of memory\n");
                return 0;
            }//(if)
        }
    }// (for(i))

```

```

    for (j=0;j<=nrows-1;j++)
        for (i=0;i<=ncolumns-1;i++)
            for (k=0;k<=nz-1;k++){
                p3[j][i][k]=0;
            }
    return (p3);
} //array3D
/*-----end array3D definitino-----*/

/*-----free alloc memory-----*/
void free1D(double* p1){
    free(p1);
    //printf("\nMemory is freed...\n");
}

void free2D(double** p2,int nx){
    for(int i=0;i<=nx-1;i++){
        free(p2[i]);
    }
    free(p2);
    //printf("\nMemory is freed...\n");
}

void free3D(double*** p3,int nx, int ny){
    for(int i=0;i<=nx-1;i++)
        for(int j=0;j<=ny-1;j++)
            free(p3[i][j]);

    for(int k=0;k<=nx-1;k++)
        free(p3[k]);
}

```

```

        //printf("\nMemory is freed...\n");
    free(p3);
}

/*-----array member operation-----*/
double array_max(double* p,int from,int to){
    double maxx=0;
    for(int i=from;i<=to;i++){
        maxx=maxx>p[i]?maxx:p[i];
    }
    return maxx;
}

double array_min(double* p,int from,int to){
    double minn=0;
    for(int i=from;i<=to;i++){
        minn=minn<p[i]?minn:p[i];
    }
    return minn;
}

double array_amp(double* p,int from,int to){
    double amp=0.0;
    for(int i=from;i<=to;i++){
        amp=amp>fabs(p[i]?amp:fabs(p[i]);
    }
    return amp;
}

/*-----FDTD Displacement vector functions-----*/
double Dx_func(double Dx_previous,double Hzxdown,double Hzxup,double hjj,

```

```

double dt){
    double Dx=0;
    Dx=Dx_previous+dt/hyj*(Hxzup-Hzxdown);
    return Dx;
}

double Dy_func(double Dy_previous,double Hzyleft,double Hzyright,double hxi,
double dt){
    double Dy=0;
    Dy=Dy_previous+dt/hxi*(Hzyleft-Hzyright);
    return Dy;
}

/*-----caculate Real and Imag parts-----*/
/*assume (x+yi)^2=(a+bi)/(c+di),and x,y unknow while a,b,c,d
given-positive root only, which means -PI<arg(z)<=PI--*/
double sqrtReal(double a,double b,double c,double d){
    double e=(a*c+b*d)/(c*c+d*d);//real part after squared
    double f=(b*c-a*d)/(c*c+d*d);//imag part after squared
    return sqrt(0.5*(e+sqrt(e*e+f*f)));
}

double sqrtImag(double a,double b,double c,double d){
    double e=(a*c+b*d)/(c*c+d*d);//real part after squared
    double f=(b*c-a*d)/(c*c+d*d);//imag part after squared
    return sqrt(0.5*(-e+sqrt(e*e+f*f)));
}

double sqrtReal(double a,double b){
    return sqrt((a+sqrt(a*a+b*b))*0.5);
}

double sqrtImag(double a,double b){
    return sqrt((-a+sqrt(a*a+b*b))*0.5);
}

```



FDTDcubic.h

```
#ifndef FDTDcubic_h
#define FDTDcubic_h
#include <complex>
using namespace std;

double square(double a);
double cube(double a);
complex<double> cbrt(complex<double>); //reload cubic root function

complex<double> cartesian(double x, double y); //complex number assign
complex<double> cartesian(complex<double> z);

class LinearFormula { //generic Linear Equation
    double _k, _b, _root; //kx+b=0;
public:
    LinearFormula(); //default constructor
    LinearFormula(double, double); //constructor
    int setFormula(double, double); //change _k, _b if necessary
    int realRootNumber();
    //find how many real roots exit, reload for 1st, 2nd and 3rd order
    double linearRoot(); //return solution
};

class QuadraticFormula { //generic Quadratic Equation,  $ax^2+bx+c=0$  with  $a=1$ ;
    double _b, _c; // $x^2+bx+c=0$ , generic form for quadratic functions
    double _delta; //discriminant,  $= b^2-4ac$  for  $a=1$ 
    complex<double> _root1, _root2;
public:
```

```

    QuadraticFormula();//default constructor

    QuadraticFormula(double ,double);// constructor

    int setFormula(double ,double);//change , _b,_c,_delta

    int realRootNumber();

    complex<double> quadraticRoot1();

    complex<double> quadraticRoot2();

};

class FullQuadraticFormula{//general Quadratic Equation , ax^2+bx+c=0
//with the option that a==0;

    double a_,b_,c_;//x^2+bx+c==0,generic form for quadratic functions

    double delta_;//discriminant , = b^2-4ac for a=1

    int isQuadratic_;//flag to show if a!=0(isQuadratic=1,true)

    complex<double> root1_,root2_;

public:

    FullQuadraticFormula();//default constructor

    FullQuadraticFormula(double ,double ,double);// constructor

    int setFormula(double ,double ,double);

    //change ,a_,b_,c_,delta_

    int realRootNumber();

    complex<double> quadraticRoot1();

    complex<double> quadraticRoot2();

};

class CubicFormula{

protected:

    double _p,_q;//x^3+ px+ q=0, canonic form

    double _discriminant;//discriminant , = (p/3)^3+(q/2)^2,

        //d<0 : 3 real roots;

        //d==0: 3 real roots and at least 2 are equal

        //d>0 : 1 real root and 2 complex roots

    complex<double> _sqrtD;

```



```

//square root of the discriminant , which is the source for complex roots
complex<double> _uu,_vv;//intemedium variables , part of _u and _v
complex<double> _u,_v;//intemedium variables , x=u+v
complex<double> CROOT2;//prefactor ,  $-1/2+i\sqrt{3}/2$ 
complex<double> CROOT3;

public:
    CubicFormula();//default constructor
    CubicFormula(double ,double);
    //constructor to initialize canonic cubic formula
    int setFormula(double ,double);
    int realRootNumber();
    complex<double> cubicRoot1();
    complex<double> cubicRoot2();
    complex<double> cubicRoot3();
};

/*Full cubicFormula*/
class FullCubicFormula{// $a_0x^3+ax^2+bx+c=0$ ;general form
protected:
    double a0_,a_;//
    double p_,q_;// $x^3+ px+ q=0$ , canonic form
    double discriminant_;//discriminant ,  $= (p/3)^3+(q/2)^2$ ,
    double isCubic_;//flag  $a_0!=0$  (true, 1)
        //d<0 : 3 real roots;
        //d==0: 3 real roots and at least 2 are equal
        //d>0 : 1 real root and 2 complex roots
    complex<double> sqrtD_;
    //square root of the discriminant , which is the source for complex roots
    complex<double> uu_,vv_;//intemedium variables , part of _u and _v
    complex<double> u_,v_;//intemedium variables , x=u+v
    complex<double> CROOT2;
    complex<double> CROOT3;

```

```
complex<double> root1_;  
complex<double> root2_;  
complex<double> root3_;  
  
public:  
    FullCubicFormula();//default constructor  
    FullCubicFormula(double ,double ,double ,double );  
    //constructor to initialize canonic cubic formula  
    int setFormula (double ,double ,double ,double );  
    int realRootNumber ();  
    complex<double> cubicRoot1 ();  
    complex<double> cubicRoot2 ();  
    complex<double> cubicRoot3 ();  
};  
#endif
```

FDTDcubic.cpp

```
#include <cmath>
#include <iostream>
#include <complex>
#include "FDTDcubic.h"
using namespace std;

double square(double a){
    return a*a;
}

double cube(double a){
    return a*a*a;
}

complex<double> cbrt(complex<double> z){
    return pow(z,1.0/3);
}

/* Cartesian Cast Function, overloaded*/
complex<double> cartesian(double x,double y){
    return complex<double> (x,y);
}

complex<double> cartesian(complex<double> z){
    return complex<double> (real(z),imag(z));
}

//kx+b==0;
/* linearFormular Class, to get the slope (root)*****/

LinearFormula::LinearFormula(){//default constructor, pascal's triangle
```

```

    _k=1;

    _b=1;

    //default solution = -1
}

LinearFormula::LinearFormula(double k,double b){

    setFormula(k,b);

}

int LinearFormula::realRootNumber(){

    if (_k!=0){

        _root=-_b/_k;

        return 1;//1 root

    }

    else{

        return 0;//no root

    }

}

double LinearFormula::linearRoot(){

    if (_k!=0){

        return (-_b/_k);

    }

    else{

        return (-1);

    }

}

int LinearFormula::setFormula(double k,double b){

    _k=k;

    _b=b;

    if (_k>0)

        return 1;

    else if(_k<0)

        return -1;

    else{

```

```

        cout<<"Warning : Slope is zero, no finite root!\n"<<endl;
        return 0;
    }
}
//x^2+bx+c==0;
/*QuadraticFormula Class, 2roots, double roots or complex roots*****/
QuadraticFormula::QuadraticFormula(){//default constructor
//using Pascal 's triangle as coefficients,which gives two identical roots -1;
    setFormula(2,1);
}
QuadraticFormula::QuadraticFormula(double b,double c){
    setFormula(b,c);
}
int QuadraticFormula::realRootNumber(){//number of real roots
    if (_delta>0)
        return 2;
    else if (_delta<0)
        return 0;
    else
        return 1;
}
int QuadraticFormula::setFormula(double b,double c){//x^2+bx+c==0
    _b=b;
    _c=c;
    _delta=_b*_b-4.0*_c;
    if (_delta>0)
        return 1;
    else if (_delta<0)
        return -1;
    else
        return 0;
}

```

```

complex<double> QuadraticFormula::quadraticRoot1(){
    if (_delta>=0)
        _root1=(-_b+sqrt(_delta))/2.0;
    else
        _root1=cartesian(-_b/2.0,sqrt(-_delta)/2.0);
    return (_root1);
}

complex<double> QuadraticFormula::quadraticRoot2(){
    if (_delta>=0)
        _root2=(-_b-sqrt(_delta))/2.0;
    else
        _root2=cartesian(-_b/2.0,-sqrt(-_delta)/2.0);
    return (_root2);
}

//ax^2+bx+c==0;
/*FullQuadraticFormula Class, 2roots, double roots or complex roots*****/
/*taking account for the case a=0*/
FullQuadraticFormula::FullQuadraticFormula(){//default constructor
//using Pascal's triangle as coefficients,which gives two identical roots -1;
    setFormula(1,2,1);
}

FullQuadraticFormula::FullQuadraticFormula(double a,double b,double c){
    setFormula(a,b,c);
}

int FullQuadraticFormula::realRootNumber(){//number of real roots
    if (delta_>0)
        return 2;
    else if (delta_<0)
        return 0;
    else
        return 1;
}

```

```

int FullQuadraticFormula::setFormula(double a,double b,double c){//ax^2+bx+c==0
    isQuadratic_=1;
    a_=a;
    if (a==0){
//        cout<<"Warning : Quadratic term vanishes!"<<endl;
        LinearFormula linearTemp(b,c);
        root1_=linearTemp.linearRoot();
        root2_=root1_;
        isQuadratic_=0;
        return -1;
    }
    else{
        b_=b/a;
        c_=c/a;
        delta_=b_*b_-4.0*c_;
        if (delta_>=0)
            return 1;
        else
            return 0;
    }
}

complex<double> FullQuadraticFormula::quadraticRoot1(){
    if(isQuadratic_!=0){
        if (delta_>=0)
            root1_=(-b_+sqrt(delta_))/2.0;
        else
            root1_=cartesian(-b_/2.0,sqrt(-delta_)/2.0);
    }
    return (root1_);
}

complex<double> FullQuadraticFormula::quadraticRoot2(){

```

```

    if(isQuadratic_!=0){
        if (delta_>=0)
            root2_=(-b_-sqrt(delta_))/2.0;
        else
            root2_=cartesian(-b_/2.0,-sqrt(-delta_)/2.0);
    }
    return (root2_);
}

/*CubicFormula Class , 3roots in total*****/
CubicFormula::CubicFormula(double p,double q){
    CROOT2=cartesian(-0.5,0.8660254037844386);//-1/2+i*sqrt(3)/2
    CROOT3=cartesian(-0.5,-0.8660254037844386);//-1/2-i*sqrt(3)/2
    setFormula(p,q);
}
int CubicFormula::setFormula(double p,double q){
    _p=p;
    _q=q;
    int returnValue=0;
    _discriminant=cube(_p/3.0)+square(_q/2.0);
    //http://en.wikipedia.org/wiki/Cubic_equation
    if (_discriminant==0){
        _sqrtD=0;
        _uu=cbrt(-_q/2.0);
        _vv=cbrt(-_q/2.0);
        returnValue=0;
    }
    if (_discriminant>0){
        _sqrtD=sqrt(_discriminant);
        _uu=cbrt(real(-_q/2.0+_sqrtD));
        _vv=cbrt(real(-_q/2.0-_sqrtD));
    }
}

```



```

        returnValue=1;
    }
    else{
        _sqrtD=cartesian(0,sqrt(-_discriminant));
        _uu=cbrt(-_q/2.0+_sqrtD);
        _vv=cbrt(-_q/2.0-_sqrtD);
        returnValue=-1;
    }
    return returnValue;
}
int CubicFormula::realRootNumber(){
    if (_discriminant<0)
        return 3;
    else
        return 1;
}
complex<double> CubicFormula::cubicRoot1(){
    _u=_uu;_v=_vv;
    return (_u+_v);
}
complex<double> CubicFormula::cubicRoot2(){
    _u=_uu*CROOT2;
    _v=_vv*CROOT3;
    return (_u+_v);
}
complex<double> CubicFormula::cubicRoot3(){
    _u=_uu*CROOT3;
    _v=_vv*CROOT2;
    return (_u+_v);
}
/*Full cubicFormula*****/

```

```

FullCubicFormula::FullCubicFormula(){
    //default, Pascal's triangle, a0 x^3+a x^2+b x+c==0;
    CROOT2=cartesian(-0.5,0.8660254037844386);//-1/2+i*sqrt(3)/2
    CROOT3=cartesian(-0.5,-0.8660254037844386);//-1/2-i*sqrt(3)/2
    setFormula(1,3,3,1);//solution x=-1 (triple)
}

FullCubicFormula::FullCubicFormula(double a0,double a,double b,double c){
    //a0 x^3+a x^2+b x+c==0;
    CROOT2=cartesian(-0.5,0.8660254037844386);//-1/2+i*sqrt(3)/2
    CROOT3=cartesian(-0.5,-0.8660254037844386);//-1/2-i*sqrt(3)/2
    setFormula(a0,a,b,c);
}

int FullCubicFormula::setFormula(double a0,double a,double b,double c){
    int returnValue=0;
    isCubic_=1;
    if (a0==0){
        isCubic_=0;//flag
        // cout<<"Warning : Cubic term vanishes!"<<endl;
        FullQuadraticFormula fullQuadraticTemp(a,b,c);
        //degenerate to x^2 formular
        root1_=fullQuadraticTemp.quadraticRoot1();
        root2_=fullQuadraticTemp.quadraticRoot2();
        root3_=root2_;
        returnValue=-1;
    }
    else{
        a/=a0;//normalize
        b/=a0;
        c/=a0;
        a_=a/3.0;//convert to canonic form
        p_=b-a*a/3.0;
    }
}

```

```

q_ = c + (2.0 * cube(a) - 9.0 * a * b) / 27.0;

discriminant_ = cube(p_ / 3.0) + square(q_ / 2.0);
//http://en.wikipedia.org/wiki/Cubic_equation
if (discriminant_ == 0){
    sqrtD_ = 0;
    uu_ = cbrt(-q_ / 2.0);
    vv_ = cbrt(-q_ / 2.0);
    returnValue = 0;
}
if (discriminant_ > 0){
    sqrtD_ = sqrt(discriminant_);
    uu_ = cbrt(real(-q_ / 2.0 + sqrtD_));
    vv_ = cbrt(real(-q_ / 2.0 - sqrtD_));
    returnValue = 1;
}
else{
    sqrtD_ = cartesian(0, sqrt(-discriminant_));
    uu_ = cbrt(-q_ / 2.0 + sqrtD_);
    vv_ = cbrt(-q_ / 2.0 - sqrtD_);
    returnValue = 3;
}
return returnValue;
}
}

int FullCubicFormula::realRootNumber(){
    if(isCubic_){
        if (discriminant_ < 0)
            return 3;
        else
            return 1;
    }
}

```

```

else
    return -1;
}
complex<double> FullCubicFormula::cubicRoot1(){
    if(isCubic_){
        u_=uu_;v_=vv_;
        return (u_+v_)-a_;
    }
    else
        return root1_;
}
complex<double> FullCubicFormula::cubicRoot2(){
    if(isCubic_){
        u_=uu_*CROOT2;
        v_=vv_*CROOT3;
        return (u_+v_)-a_;
    }
    else
        return root2_;
}
complex<double> FullCubicFormula::cubicRoot3(){
    if(isCubic_){
        u_=uu_*CROOT3;
        v_=vv_*CROOT2;
        return (u_+v_)-a_;
    }
    else
        return root3_;
}

```

FDTDtime.h

```
#ifndef FDTDtime_h
#define FDTDtime_h
/*Estimate finishing time of the simulation program*/
int FDTDtimer(int total_I,int current_i,int inc_i,clock_t previous_clock);

#endif
```

FDTDtime.cpp

```
#include "FDTDarray.h"
#include <ctime>
#include <iostream>

using namespace std;
/*-----Timer and/or estimated time function-----*/
int FDTDtimer(int total_I,int current_i,int inc_i,clock_t previous_clock){
    int i=current_i;
    int totalII=total_I;
    time_t ptimer2;
    time_t ptimer3;

    clock_t clock1=previous_clock;
    clock_t clock2=clock();//current clock
    clock_t clock3;
    clock_t clock_diff;
    clock_t clock_total;

    struct tm *timeinfo2;
    struct tm *timeinfo3;
    int second_total;
    int second_frac;
    int hour_total;
    int minute_total;

    cout<<endl;
    //cout<<"i="<<i<<endl;
```

```

time(&ptimer2);
//cout<<"ptimer2==="<<ptimer2<<endl;
//cout<<"clock1==="<<clock1<<endl;

//cout<<"clock2==="<<clock2<<endl;
clock_diff=clock2-clock1;

clock_total=(int)(totalII-(double)i)/inc_i*clock_diff;
//cout<<"clock_total tick remaining...(clock_t)"<<clock_total<<endl;

second_total=clock_total/CLOCKS_PER_SEC;
second_frac=second_total;
//cout<<"second_total remaining...(sec)"<<second_total<<endl;
minute_total=second_frac/60;
second_frac-=minute_total*60;
hour_total=minute_total/60;
minute_total-=hour_total*60;

cout<<"Total time estimated: "<<hour_total<<" hours , "<<
minute_total<<" mintes and "<<second_frac<<" seconds~"<<endl;

//cout<<"clock loop diff (sec)"<<((double)clock_diff/CLOCKS_PER_SEC<<endl;
//cout<<"current time (time_t)"<<ptimer2<<endl;

//cout<<endl;
clock3=clock2+clock_total;
//cout<<"clock now (clock_t)"<<clock()<<endl;
//cout<<"finishing time (clock_t)"<<clock3<<endl;

//cout<<endl;
ptimer3=second_total+ptimer2;
//cout<<"finishing timer (time_t)"<<ptimer3<<endl;

```

```
timeinfo2=localtime(&ptimer2);  
//cout<<"current asctime===="<<asctime(timeinfo2)<<endl;  
timeinfo3=localtime(&ptimer3);  
cout<<"Finishing asctime : \t"<<asctime(timeinfo3)<<endl;  
  
cout<<endl<<endl;  
return 0;  
  
}
```


REFERENCES CITED

- [1] A. B. Kahng, Design Test of Computers, IEEE **27**, 86 (2010), ISSN 0740-7475.
- [2] T.-Y. Chiang, B. Shieh, and K. C. Saraswat, IEEE Symp. VLSI Circuits, Dig. Tech pp. 38–39 (2002).
- [3] E. Ozbay, Science **311**, 189 (2006), <http://www.sciencemag.org/content/311/5758/189.full.pdf>, URL <http://www.sciencemag.org/content/311/5758/189.abstract>.
- [4] R. Zia, J. Schuller, A. Chandran, and M. Brongersma, Materials Today **9**, 20 (2006), ISSN 13697021, URL [http://dx.doi.org/10.1016/S1369-7021\(06\)71572-3](http://dx.doi.org/10.1016/S1369-7021(06)71572-3).
- [5] I. P. Kaminow, T. Li, and A. E. Willner, *Optical fiber Telecommunications V A: Components and Subsystems, Volume 1* (Academic Press, 2008).
- [6] D. Pines, Rev. Mod. Phys. **28**, 184 (1956).
- [7] U. Fano, Phys. Rev. **103**, 1202 (1956).
- [8] S. Cunningham, A. Maradudin, and R. Wallis, Phys. Rev. B **10** (1974).
- [9] P. Berini, R. Charbonneau, N. Lahoud, and G. Mattiussi, **98**, 043109 (2005), ISSN 00218979, URL <http://dx.doi.org/doi/10.1063/1.2008385>.
- [10] W. L. Barnes, A. Dereux, and T. W. Ebbesen, Nature **424** (2003).
- [11] M. L. Brongersma, J. W. Hartman, and H. A. Atwater, Phys. Rev. B **62**, R16356 (2000), URL <http://link.aps.org/doi/10.1103/PhysRevB.62.R16356>.
- [12] K. Hasegawa, J. Nöckel, and M. Deutsch, Appl. Phys. Lett. **84**, 1835 (2004).
- [13] F. Yang, J. R. Sambles, and G. W. Bradberry, Phys. Rev. Lett. **64**, 559 (1990), URL <http://link.aps.org/doi/10.1103/PhysRevLett.64.559>.

- [14] F. Yang, J. R. Sambles, and G. W. Bradberry, *Phys. Rev. B* **44**, 5855 (1991), URL <http://link.aps.org/doi/10.1103/PhysRevB.44.5855>.
- [15] S. I. Bozhevolnyi and F. A. Pudonin, *Phys. Rev. Lett.* **78**, 2823 (1997), URL <http://link.aps.org/doi/10.1103/PhysRevLett.78.2823>.
- [16] T. Nikolajsen, K. Leosson, and S. I. Bozhevolnyi, *Applied Physics Letters* **85** (2004).
- [17] K. Hasegawa, J. Nöckel, and M. Deutsch, *Phys. Rev. A* **75**, 063816 (2008).
- [18] M. Willatzen, *Phys. Rev. A* **80**, 043805 (2009), URL <http://link.aps.org/doi/10.1103/PhysRevA.80.043805>.
- [19] S. A. Maier, M. L. Brongersma, P. G. Kik, S. Meltzer, A. A. G. Requicha, and H. A. Atwater, *Advanced Materials* **13**, 1501 (2001), ISSN 1521-4095, URL [http://dx.doi.org/10.1002/1521-4095\(200110\)13:19<1501::AID-ADMA1501>3.0.CO;2-Z](http://dx.doi.org/10.1002/1521-4095(200110)13:19<1501::AID-ADMA1501>3.0.CO;2-Z).
- [20] J. A. Dionne, L. A. Sweatlock, H. A. Atwater, and A. Polman, *Phys. Rev. B* **72**, 075405 (2005), URL <http://link.aps.org/doi/10.1103/PhysRevB.72.075405>.
- [21] J. A. Dionne, K. Diest, L. A. Sweatlock, and H. A. Atwater, *Nano Letters* **9**, 897 (2009), pMID: 19170558, <http://pubs.acs.org/doi/pdf/10.1021/nl803868k>, URL <http://pubs.acs.org/doi/abs/10.1021/nl803868k>.
- [22] K. F. MacDonald, Z. L. Sámson, M. I. Stockman, and N. I. Zheludev, *Nature Photonics* **3**, 55 (2009), URL <http://www.nature.com/nphoton/journal/v3/n1/pdf/nphoton.2008.249.pdf>.
- [23] N. Rotenberg, J. N. Caspers, and H. M. van Driel, *Phys. Rev. B* **80**, 245420 (2009), URL <http://link.aps.org/doi/10.1103/PhysRevB.80.245420>.
- [24] Z. L. Sámson, K. F. MacDonald, and N. I. Zheludev, *Journal of Optics A: Pure and Applied Optics* **11**, 114031 (2009), URL <http://stacks.iop.org/1464-4258/11/i=11/a=114031>.
- [25] M. Piliarik, L. Párová, and J. Homola, *Biosensors and Bioelectronics* **24** (2009).
- [26] C. Nylander, B. Liedberg, and T. Lind, *Sensors and Actuators* **3**, 79 (1982).
- [27] B. Liedberg, C. Nylander, and I. Lundström, *Sensors and Actuators* **4**, 299 (1983).
- [28] J. Homola, S. S. Yee, and G. Gauglitz, *Sensors and Actuators B* **54**, 3 (1999).

- [29] B. Liedberg, C. Nylander, and I. Lundström, *Biosensors Bioelectron* **10**, i (1995).
- [30] M. A. Cooper, *Nature* **1** (2002).
- [31] R. Harris and J. Wilkinson, *Sensors and Actuators B* **29**, 261 (1995).
- [32] S. Ushioda and Y. Sasaki, *Phys. Rev. B* **27**, 1401 (1983), URL <http://link.aps.org/doi/10.1103/PhysRevB.27.1401>.
- [33] A. Otto, I. Mrozek, H. Grabhorn, and W. Akemann, *Journal of Physics: Condensed Matter* **4**, 1143 (1992), URL <http://stacks.iop.org/0953-8984/4/i=5/a=001>.
- [34] A. C. R. Pipino, G. C. Schatz, and R. P. Van Duyne, *Phys. Rev. B* **49**, 8320 (1994), URL <http://link.aps.org/doi/10.1103/PhysRevB.49.8320>.
- [35] K. A. O'Donnell, R. Torre, and C. S. West, *Phys. Rev. B* **55**, 7985 (1997), URL <http://link.aps.org/doi/10.1103/PhysRevB.55.7985>.
- [36] L. Cao, N. C. Panoiu, and R. M. Osgood, *Phys. Rev. B* **75**, 205401 (2007), URL <http://link.aps.org/doi/10.1103/PhysRevB.75.205401>.
- [37] E. D. Palik, *Handbook of Optical Constants of Solids* (Academic Press, 1985).
- [38] J. Jackson, *Classical Electrodynamics* (John Wiley & Sons, Inc., 1998).
- [39] N. Bloembergen, *Nonlinear Optics* (World Scientific Publishing Co Pte Ltd, 1996).
- [40] Y.-R. Shen, *The Principles of Nonlinear Optics* (John Wiley & Sons, 1984).
- [41] R. W. Boyd, *Nonlinear Optics* (Academic Press, 2008).
- [42] Y. Guo and M. Deutsch, *Frontiers in Optics (FiO)/Laser Science XXVI (LS) conference* **696** (2010).
- [43] Y. Guo and M. Deutsch, *APS March Meeting* (2010).
- [44] S. Maier and H. A. Atwater, *Journal of applied physics* **98** (2005).
- [45] P. Berini, *Optics Letters* **24** (1999).
- [46] N. W. Ashcroft and N. D. Mermin, *Solid State Physics* (Brooks Cole, 1976).
- [47] S. Maier, *Plasmonics - Fundamentals and Applications* (Springer, 2007).

- [48] M. J. Keskinen, P. Loschialpo, D. Forester, and J. Schelleng, **88**, 5785 (2000), ISSN 00218979, URL <http://dx.doi.org/10.1063/1.1289045>.
- [49] H. S. Nalwa, *Handbook of Advanced Electronic and Photonic Materials and Devices: Conducting polymers* (Elsevier, 2001).
- [50] K. Lee, R. Menon, C. O. Yoon, and A. J. Heeger, Phys. Rev. B **52**, 4779 (1995), URL <http://link.aps.org/doi/10.1103/PhysRevB.52.4779>.
- [51] K. Lee and A. J. Heeger, Phys. Rev. B **68**, 035201 (2003), URL <http://link.aps.org/doi/10.1103/PhysRevB.68.035201>.
- [52] S. R. Nagel and S. E. Schnatterly, Phys. Rev. B **9**, 1299 (1974), URL <http://link.aps.org/doi/10.1103/PhysRevB.9.1299>.
- [53] P. G. Etchegoin, E. C. L. Ru, and M. Meyer, **125**, 164705 (2006), ISSN 00219606, URL <http://dx.doi.org/doi/10.1063/1.2360270>.
- [54] I. R. Hooper and J. R. Sambles, Phys. Rev. B **65**, 165432 (2002).
- [55] D. Rogovin and T. P. Shen, Phys. Rev. B **37**, 1121 (1988), URL <http://link.aps.org/doi/10.1103/PhysRevB.37.1121>.
- [56] S. Palomba and L. Novotny, Phys. Rev. Lett. **101**, 056802 (2008), URL <http://link.aps.org/doi/10.1103/PhysRevLett.101.056802>.
- [57] C. Tang, Z. Wang, W. Zhang, S. Zhu, N. Ming, G. Sun, and P. Sheng, Phys. Rev. B **80**, 165401 (2009), URL <http://link.aps.org/doi/10.1103/PhysRevB.80.165401>.
- [58] P. A. Franken, A. E. Hill, C. W. Peters, and G. Weinreich, Phys. Rev. Lett. **7**, 118 (1961), URL <http://link.aps.org/doi/10.1103/PhysRevLett.7.118>.
- [59] F. Moulin and D. Bernard, Optics Communications **164**, 137 (1999), ISSN 0030-4018, URL <http://www.sciencedirect.com/science/article/pii/S0030401899001698>.
- [60] D. F. Eaton, Science **253**, 281 (1991).
- [61] M. J. Comstock, *Materials for Nonlinear Optics* (American Chemical Society, 1991).
- [62] J. Zyss, *Molecular nonlinear optics: Materials, physics and devices* (Academic Press, 1997).
- [63] H. S. Nalwa and S. Miyata, *Nonlinear optics of organic molecules and polymers* (CRC Press, 1997).

- [64] N. Bloembergen, Applied Physics B **68** (1999).
- [65] J. E. Sipe, V. C. Y. So, M. Fukui, and G. I. Stegeman, Phys. Rev. B **21**, 4389 (1980), URL <http://link.aps.org/doi/10.1103/PhysRevB.21.4389>.
- [66] F. De Martini, P. Ristori, E. Santamato, and A. C. A. Zammit, Phys. Rev. B **23**, 3797 (1981), URL <http://link.aps.org/doi/10.1103/PhysRevB.23.3797>.
- [67] J. Renger, R. Quidant, N. van Hulst, S. Palomba, and L. Novotny, Phys. Rev. Lett. **103**, 266802 (2009), URL <http://link.aps.org/doi/10.1103/PhysRevLett.103.266802>.
- [68] D. Moss, J. E. Sipe, and H. M. van Driel, Physical Review B **35**, 1130 (1986).
- [69] G. Lüpke and G. Marowsky, Applied Physical B **53**, 71 (1991).
- [70] H. J. Simon, D. E. Mitchell, and J. G. Watson, Phys. Rev. Lett. **33**, 1531 (1974), URL <http://link.aps.org/doi/10.1103/PhysRevLett.33.1531>.
- [71] C.-C. Tzeng and J. T. Lue, Phys. Rev. A **39**, 191 (1989), URL <http://link.aps.org/doi/10.1103/PhysRevA.39.191>.
- [72] G. S. Agarwal and S. S. Jha, Phys. Rev. B **26**, 482 (1982), URL <http://link.aps.org/doi/10.1103/PhysRevB.26.482>.
- [73] M. G. Weber, Phys. Rev. B **33**, 6775 (1986), URL <http://link.aps.org/doi/10.1103/PhysRevB.33.6775>.
- [74] J. Nkoma, R. Loudon, and D. R. Tilley, J. Phys. C: Solid State Phys. **7**, 3547 (1974).
- [75] H. E. J. Dodd, R.K.and Morris, *Soliton and nonlinear wave equations* (Academic Press, 1982).
- [76] N. J. Zabusky and M. D. Kruskal, Phys. Rev. Lett. **15**, 240 (1965), URL <http://link.aps.org/doi/10.1103/PhysRevLett.15.240>.
- [77] C. S. Gardner, J. M. Greene, M. D. Kruskal, and R. M. Miura, Phys. Rev. Lett. **19**, 1095 (1967), URL <http://link.aps.org/doi/10.1103/PhysRevLett.19.1095>.
- [78] H. C. Yuen and B. M. Lake, **18**, 956 (1975), ISSN 00319171, URL <http://dx.doi.org/doi/10.1063/1.861268>.
- [79] M. Destrade and G. Saccomandi, Phys. Rev. E **73**, 065604 (2006), URL <http://link.aps.org/doi/10.1103/PhysRevE.73.065604>.

- [80] S. Bednarek, B. Szafran, and K. Lis, Phys. Rev. B **72**, 075319 (2005), URL <http://link.aps.org/doi/10.1103/PhysRevB.72.075319>.
- [81] W. Singhsomroje and H. J. Maris, Phys. Rev. B **69**, 174303 (2004), URL <http://link.aps.org/doi/10.1103/PhysRevB.69.174303>.
- [82] I. V. Shadrivov, A. A. Sukhorukov, Y. S. Kivshar, A. A. Zharov, A. D. Boardman, and P. Egan, Phys. Rev. E **69**, 016617 (2004), URL <http://link.aps.org/doi/10.1103/PhysRevE.69.016617>.
- [83] A. R. Davoyan, I. V. Shadrivov, and Y. S. Kivshar, Opt. Express **17**, 21732 (2009), URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-17-24-21732>.
- [84] L. F. Mollenauer, R. H. Stolen, and J. P. Gordon, Phys. Rev. Lett. **45**, 1095 (1980), URL <http://link.aps.org/doi/10.1103/PhysRevLett.45.1095>.
- [85] A. Hasegawa and F. Tappert, Applied Physics Letters **23**, 142 (1973), URL <http://link.aip.org/link/?APL/23/142/1>.
- [86] Y. S. Kivshar and G. P. Agrawal, *Optical Solitons* (Academic Press, 2003).
- [87] R. H. Stolen and C. Lin, Phys. Rev. A **17**, 1448 (1978), URL <http://link.aps.org/doi/10.1103/PhysRevA.17.1448>.
- [88] G. I. Stegeman and M. Segev, Science **286**, 1518 (1999), URL <http://www.sciencemag.org/content/286/5444/1518.full>.
- [89] G. Stegeman, D. Christodoulides, and M. Segev, Selected Topics in Quantum Electronics, IEEE Journal of **6**, 1419 (2000), ISSN 1077-260X.
- [90] D. Korteweg and G. de Vries, Philos. Mag. **39** (1895).
- [91] J. R. Taylor, *Optical Solitons - Theory and Experiment* (Cambridge University Press, 1992).
- [92] F. Abdullaev, S. Darmanyan, and P. Khabibullaev, *Optical Solitons* (Academic Press, 1982).
- [93] A. D. Boardman, G. S. Cooper, A. A. Maradudin, and T. P. Shen, Physical review B **34**, 8273 (1986).
- [94] G. Agrawal, *Nonlinear Fiber Optics* (Academic Press, 2006).
- [95] M. Kyoung and M. Lee, Optics Communications **171**, 145 (1999), ISSN 0030-4018, URL <http://www.sciencedirect.com/science/article/pii/S0030401899005519>.

- [96] R. del Coso and J. Solis, *J. Opt. Soc. Am. B* **21**, 640 (2004), URL <http://josab.osa.org/abstract.cfm?URI=josab-21-3-640>.
- [97] J.-H. Huang, R. Chang, P.-T. Leung, and D. P. Tsai, *Optics Communications* **282**, 1412 (2009), ISSN 0030-4018, URL <http://www.sciencedirect.com/science/article/pii/S0030401808012480>.
- [98] D. Mihalache, G. I. Stegeman, C. T. Seaton, E. M. Wright, R. Zaroni, A. D. Boardman, and T. Twardowski, *Opt. Lett.* **12**, 187 (1987), URL <http://ol.osa.org/abstract.cfm?URI=ol-12-3-187>.
- [99] S. A. Ramakrishna and O. J. Martin, *Opt. Lett.* **30**, 2626 (2005), URL <http://ol.osa.org/abstract.cfm?URI=ol-30-19-2626>.
- [100] A. M. Weiner, J. P. Heritage, R. J. Hawkins, R. N. Thurston, E. M. Kirschner, D. E. Leaird, and W. J. Tomlinson, *Phys. Rev. Lett.* **61**, 2445 (1988), URL <http://link.aps.org/doi/10.1103/PhysRevLett.61.2445>.
- [101] Y. S. Kivshar and B. A. Malomed, *Rev. Mod. Phys.* **61**, 763 (1989), URL <http://link.aps.org/doi/10.1103/RevModPhys.61.763>.
- [102] O. Katz, Y. Lahini, and Y. Silberberg, *Optical Letters* **33** (2008).
- [103] K.-J. Boller, A. Imamolu, and S. E. Harris, *Phys. Rev. Lett.* **66**, 2593 (1991), URL <http://link.aps.org/doi/10.1103/PhysRevLett.66.2593>.
- [104] A. Kumar, S. F. Yu, X. F. Li, and S. P. Lau, *Opt. Express* **16**, 16113 (2008), URL <http://www.opticsexpress.org/abstract.cfm?URI=oe-16-20-16113>.
- [105] K. Hasegawa, C. Rohde, and M. Deutsch, *Optics Letters* **31** (2006).
- [106] G. Veronis and S. Fan, *J. Lightwave Technol.* **25**, 2511 (2007), URL <http://jlt.osa.org/abstract.cfm?URI=jlt-25-9-2511>.
- [107] M. I. Stockman, *Phys. Rev. Lett.* **106**, 156802 (2011), URL <http://link.aps.org/doi/10.1103/PhysRevLett.106.156802>.
- [108] E. Kreyszig, *Advanced Engineering Mathematics* (John Wiley & Sons, Inc, 1998).
- [109] G. B. Arfken, H. J. Weber, and F. Harris, *Mathematical Methods for Physicists, Sixth Edition: A Comprehensive Guide* (Academic Press, 2005).
- [110] T. R. Taha and M. J. Ablowitz, *Journal of Computational Physics* **55**, 203 (1984).

- [111] C. Pozrikidis, *Numerical Computation in Science and Engineering* (Oxford University Press, 1998).
- [112] D.-C. Ionescu and H. Igel, *Lecture notes in computer science* **2659**, 807 (2003).