

DDSURF User's Manual

CREATETAR PC Version 2007

DDSURF PC Version 2007

FFSURF PC Version 2007

March 2007

Brent M. Nebeker

Haiping Zhang

Euiwon Bae

E. Dan Hirleman

Purdue University

Supported by:

The Consortium for Metrology of Semiconductor Nanodefects,  
Arizona State University,  
and Purdue University

## Preface

DDSURF is a group of electromagnetic scatter modeling software codes that have been under development since 1994 at the Laser Diagnostics Laboratory of Arizona State University as well as Purdue University since 1999. This project is currently under the sponsorship of the Consortium for Metrology of Semiconductor Nanodefects, Arizona State University, and Purdue University, and has been under the sponsorship at various times by SEMATECH and the Semiconductor Research Corporation. The three main codes of DDSURF are CREATETAR (*Create Target*), DDSURF (*Near-Field*), and FFSURF (*Far-Field*). CREATETAR is a pre-processor which creates the geometry of the elements used in the modeling of the scattering feature. DDSURF uses the geometry file created by CREATETAR and models the internal electromagnetic response of a feature in free space or on or near a surface when illuminated by an electromagnetic source. Using the internal field computed by DDSURF, FFSURF computes the far-field response of the scatterer. Development of DDSURF was inspired by the code DDSCAT, developed by B. T. Draine and P. J. Flatau (1994). DDSCAT models electromagnetic scattering from features in free space. To model the electromagnetic scattering from features near surfaces, DDSURF uses portions of the SOMNEC code developed at the Lawrence Livermore Laboratory, created as a preprocessor for NEC (*Numerical Electromagnetics Code*) (Burke, 1981). This User's Manual is the for the fourth DDSURF version released to the members of the Consortium for Metrology of Semiconductor Nanodefects. The previous software releases were in January 1997 (DDSURF 1-1997), January 1998 (DDSURF 1-1998), January 1999 (DDSURF 1-1999), April 1999 (DDSURF 2-1999), and January 2000 (DDSURF 1-2000).

The present version of DDSURF was developed using Digital Visual FORTRAN 90 and is the result of efforts by Brent M. Nebeker, Haiping Zhang, Euiwon Bae and E. Dan Hirleman of Purdue University, and Roland Schmehl of the University of Karlsruhe, Germany.

## Abstract

DDSURF is a group of software codes developed to model electromagnetic light scattering from small, arbitrarily shaped features on or near flat surfaces. These codes were developed especially to be used as a tool for the semiconductor industry where optical detection and characterization of structure defects and particle contamination on silicon wafers is essential to improve production yield. The three main codes of DDSURF are CREATETAR (*Create Target*), DDSURF (*Internal Dipole Moment*) and FFSURF (*Far-Field*). The scattering response of an illuminated feature is modeled by using the discrete-dipole approximation method (DDA). In this method discrete electric dipoles are used to simulate the electromagnetic response of a feature. CREATETAR creates that lattice array where dipoles are positioned to model a feature geometry. By the numerical solution of the interaction equations of electromagnetically excited dipoles, DDSURF computes the dipole moment distribution within the scattering feature using the array created by CREATETAR. Using the dipole moment distribution found by DDSURF, FFSURF computes the external scattered far-field by applying the Green's function for a radiating source.

The DDSURF User's Manual contains the theoretical background of the codes including description of the DDA method, and interaction between the surface and the dipoles. General equations describing the dipole interaction are provided. Instructions for use of the codes are given in the manual, including determination of the input parameters and the explanation of the output files created by these codes.

## Table of Contents

<b>PREFACE</b> .....	<b>2</b>
<b>ABSTRACT</b> .....	<b>3</b>
<b>TABLE OF CONTENTS</b> .....	<b>4</b>
<b>I INTRODUCTION</b> .....	<b>6</b>
<b>II THEORY</b> .....	<b>6</b>
II.1 THE DISCRETE-DIPOLE APPROXIMATION METHOD (COMPUTATION OF THE INTERNAL FIELD OF THE SCATTERING FEATURE) .....	6
II.2 COMPUTATION OF THE EXTERNAL FIELD .....	10
<b>III. COMPARISON WITH EXPERIMENT</b> .....	<b>12</b>
III.1 COMPLEX SHAPES ON SURFACE.....	12
III.2 FEATURES ON FILMED SURFACES .....	14
<b>IV. DDSURF 2007-1 SOFTWARE</b> .....	<b>16</b>
IV.1 CREATETAR.....	16
IV.1.1 CREATETAR Input Parameters.....	16
IV.1.2 Execution of CREATETAR.....	17
IV.1.3 SHAPE.DAT.....	18
IV.1.4 Example Feature Shape Designations (Pre-defined Shapes).....	18
IV.1.4 Example Feature Shape Designations (User Defined Shapes).....	26
IV.2 DDSURF .....	28
IV.2.1 DDSURF – Input Parameters.....	28
IV.2.2 Execution of DDSURF.....	32
IV.2.3 Output files of FFSURF .....	32
IV.3 FFSURF.....	33
IV.3.1 FFSURF – Input Parameters.....	33
IV.3.2 Execution of FFSURF.....	36
IV.3.3 Output files of FFSURF .....	36
<b>V REFERENCES</b> .....	<b>38</b>
<b>APPENDIX A</b> .....	<b>40</b>
A.1 DDSURF – FILE LIST .....	40
A.2 CREATETAR - CODE SETUP AND SUBROUTINE LIST .....	40
A.2.1 Block Diagram of CREATETAR .....	40
A.3 DDSURF - CODE BLOCK DIAGRAM AND SUBROUTINE LIST .....	40
A.3.1 Block Diagram of DDSURF.....	40
A.4 FFSURF - CODE BLOCK DIAGRAM AND SUBROUTINE LIST .....	41
A.4.1 Block Diagram of FFSURF.....	41
<b>APPENDIX B</b> .....	<b>42</b>
B.1 DYADIC GREEN’S FUNCTION .....	42
B.2 REFLECTION COEFFICIENTS .....	42
B.3 SOMMERFELD INTEGRAL TERMS.....	42
B.4 GAUSSIAN BEAM EQUATIONS .....	43
B.5 POLARIZABILITY MODELING - LATTICE DISPERSION RELATION AND ELLIPTICAL DIPOLE MODELING .....	44

<b>APPENDIX C .....</b>	<b>47</b>
C.1 CREATETAR .....	47
C.1.1 CREATETAR Execution.....	47
C.2 DDSURF .....	47
C.2.1 DDSURF Execution.....	47
C.2.2 DDSURF output files .....	49
C.3 FFSURF .....	51
C.3.1 Field Distribution along a plane detection region of constant $x$ .....	51
C.3.2 Field Distribution along a hemispherical region of constant radius.....	57
C.3.3 Field Distribution along plane of incidence at constant radius.....	63

## I INTRODUCTION

The software DDSURF for a Windows based PC system is a group of codes used to predict electromagnetic scattering from illuminated features present either in free space, on top of a flat surface, or suspended above a flat surface. DDSURF consists of three main executable codes: (1) *createtar07.exe*— which configures the dipole lattice array used to model the geometry of the feature; (2) *ddsurf07.exe* - which calculates the internal field (dipole moment distribution) within the scattering feature; and (3) *ffsurf07.exe*- which uses the internal field (dipole moment distribution) to determine the external far-field quantities of the scattered field. These quantities include the scattered irradiance and electromagnetic field distribution, and the total scattering cross-section.

DDSURF is based on the *Discrete-Dipole Approximation* method (DDA), also referred to as the coupled-dipole approximation method. The DDA uses the excitation response of electromagnetic dipoles to model the electromagnetic scattering characteristics of a feature. The dipoles are placed in a lattice configuration which models the physical geometries of the scattering feature. CREATETAR creates the lattice configuration used by DDSURF. Using an iterative process to solve the interaction equations of electromagnetically excited dipoles, DDSURF computes the feature internal dipole moment distribution. FFSURF computes the external scattered field by applying the far-field Green's function to the dipole moment distribution found by DDSURF.

In the DDSURF User's Manual, we first discuss the theoretical background of the DDA method used in the development of the codes. Next we discuss the process in which DDSURF was validated, showing comparisons with experiments that were conducted at Arizona State University. Next, we discuss the codes of DDSURF, first how to use the pre-processor CREATETAR to develop the scattering feature dipole lattice configuration. Next, we discuss the DDSURF code itself, what input parameters are needed, and what the steps are needed to run the code. The aspects of the output data files of DDSURF are discussed next. Then the post-processor FFSURF is discussed next, and what parameters will be needed for that code, and what output files can be created with its execution. There are a variety of options to evaluate the results of FFSURF. For example, the scattered irradiance distributions can be plotted, as well as the differential scattering cross-section. Appendix C shows examples of the output files of DDSURF and FFSURF.

## II THEORY

### II.1 The Discrete-Dipole Approximation Method (Computation of the Internal Field of the Scattering Feature)

Starting from the molecular level, we will discuss the theory behind the discrete-dipole approximation (DDA) method used to model the electromagnetic scattering from features. Molecules are usually electrically neutral, but they are made of atoms which contain positively charge nuclei and negatively charged electrons. When an electric field is present on an atom, the nucleus moves a small distance in the direction of the field while the electrons move in the opposite direction, thus creating an oscillating *dipole*. Figure 2.1 shows an example of a dipole:

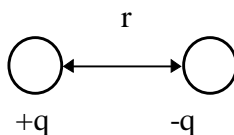


Figure 2.1 Example of an electric dipole.

Similar to the magnetic moment and field created by a magnetic bar with two different poles, the result of separation of the two charges by a distance is a dipole moment and an electric field.

The discrete-dipole approximation method (DDA) was first developed by Purcell and Pennypacker (1973) and later studied by other researchers such as Draine (1988) and Taubenblatt and Tran (1993). In the DDA, sub volumes of features subjected to an electric field are modeled as discrete dipoles. Figure 2.2 shows an example of a dipole configuration for a scattering feature on a surface.

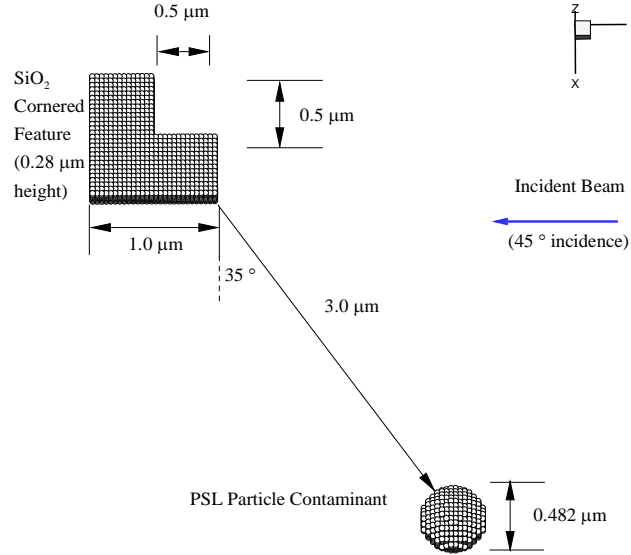


Figure 2.2 Example of dipole configuration for a cornered feature with a spherical particle contaminant on a surface.

The electromagnetic response of each dipole can be determined mathematically by the dipole equations. The dipole moment at the dipole is related to the incident electric field by:

$$\overline{P}_i = \alpha_i \overline{E}_{tot,i} \quad (2-1)$$

$\overline{P}_i$  is the dipole moment at dipole  $i$ ,  $\overline{E}_{tot,i}$  is the total electric field at dipole  $i$ , and  $\alpha_i$  is the dipole polarizability. The polarizability is a function of the dipole size and feature material. There has been extensive work to correctly model the polarizability, thus several methods are available. DDSURF uses the lattice dispersion relation (LDR), developed by Draine and Goodman (1993), and has extended this method to model non-spherical dipoles.

The field at a dipole,  $\overline{E}_{tot,i}$ , can be characterized into several modes. These modes include the direct field from the electrical field incident upon the features and the electric field contribution from the other interacting dipoles. There are two parts to each of these modes, the direct field, and the field when the interacting surface is involved

The electric field can be represented in equation form as:

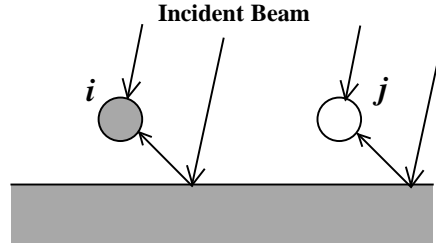
$$\overline{E}_{tot,i} = \overline{E}_{o,i} + \overline{E}_{direct,i} + \overline{E}_{reflected,i} \quad (2-2)$$

Substituting this into Eq. (2-1), we can make the dipole equation into the form:

$$(\alpha_i)^{-1} \overline{P}_i - \overline{E}_{direct,i} - \overline{E}_{reflected,i} = \overline{E}_{o,i} \quad (2-3)$$

We will now discuss the three components of the electric field at a dipole.

#### Electric Field by Incident Beam



The field due to the light beam incident on the feature has two components, the direct beam, and the field that was reflected from the surface. The reflected field can be computed using the Fresnel reflection equations. Thus the field from the incident beam,  $\bar{E}_{0,i}$ , has the following form in the  $x$ ,  $y$ , and  $z$  components:

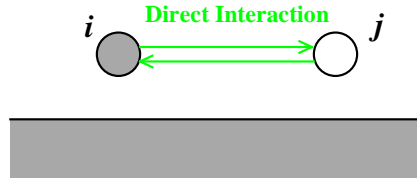
$$E_x = E_{0x} \left( e^{-ik_z z} - R^{TM} e^{ik_z z} \right) e^{ik_x x} \quad (2-4a)$$

$$E_y = E_{0y} \left( e^{-ik_z z} + R^{TE} e^{ik_z z} \right) e^{ik_x x} \quad (2-4b)$$

$$E_z = E_{0z} \left( e^{-ik_z z} + R^{TE} e^{ik_z z} \right) e^{ik_x x} \quad (2-4c)$$

The reflection coefficients are described in Appendix B.

#### Electric Field by Direct Dipole Interaction

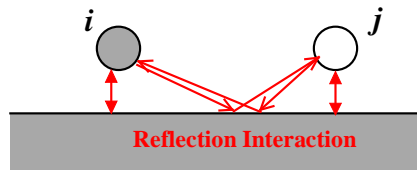


The DDSCAT code developed by Draine and Flatau (1994) considered only direct interaction between dipoles, that is, there was no reflection interaction with a surface. The electric field at a particular dipole,  $i$ , due to the field emitted by the other dipoles,  $j$ , in the scattering feature,  $\bar{E}_{direct,i}$ , can be computed by the summation:

$$\bar{E}_{direct,i} = \frac{k_0^2}{\epsilon_0} \sum_{j \neq i} \bar{G}_{ij} \cdot \bar{P}_j \quad (2-5)$$

Where  $k_0$  is the wave number in free space, and  $\epsilon_0$  is the permittivity of free space. The Green's function,  $\bar{G}_{ij}$  is defined in Appendix B.

#### Electric Field by Reflected Dipole Interaction



A unique component of DDSURF is the modeling of the dipole radiation that is reflected at the surface. The dipole interaction with the surface occurs in the near-field zone, thus when the electric field interacts with the surface, it does not act as a plane wave. Thus the sole use of Fresnel coefficients to determine the reflected fields is not an accurate model. To deal with the near-field effects, the Sommerfeld integrals are used.



A derivation to determine for the electric field for a dipole above a surface is given by Baños (Baños, 1966). Baños separates the radiating electric field into two components,  $E_\rho$ , the electric field in the plane parallel to the surface ( $x$  and  $y$  directions), and  $E_z$ , the electric field in the plane perpendicular to the surface ( $z$  direction.)

$$E_\rho(r) = \frac{p}{\epsilon_1} \left\{ \frac{\partial^2}{\partial \rho \partial z} \frac{e^{ik_1 r}}{4\pi r} \right\} \quad (2-6a)$$

$$E_z(r) = \frac{p}{\epsilon_0} \left\{ \left( k_2^2 + \frac{\partial^2}{\partial z^2} \right) \frac{e^{ik_2 r}}{4\pi r} \right\} \quad (2-6b)$$

Through the years, much work has been conducted to evaluate these equations. In particular, the Sommerfeld integral relation states:

$$\frac{e^{ikr}}{r} = i \int_0^\infty \frac{k_\rho}{k_z} J_0(k_\rho \rho) e^{ik_z |z|} dk_\rho \quad (2-7)$$

There has been extensive investigation on numerical methods to integrate this integral (Lager *et al.*, 1974; Mohsen, 1982). DDSURF uses routines developed by Lytle for evaluating the Sommerfeld integral relation. With the use of these routines, the reflected component of the electric field due to a radiating dipole in the near field can be represented as:

$$\bar{E}_{reflected,i} = \sum_{j=1}^N \left( \bar{S}_{ij} + \frac{k_2^2}{\epsilon_1} \frac{k_1^2 - k_2^2}{k_1^2 + k_2^2} \bar{G}_{ij}^I \right) \cdot \bar{P}_j. \quad (2-8)$$

$\bar{S}_{ij}$  is a 3 x 3 matrix which are Sommerfeld integral expressions, which are determined by the routines provided by Lager (Lager *et al.*, 1974), where  $k_1$  and  $k_2$  are the wave numbers for the particle and the surface respectively, and  $\epsilon_1$  is the permittivity. For the form of  $\bar{S}_{ij}$ , refer to Schmehl et al. (1997). The image dyadic Green's function is defined as (Schmehl, 1994)

$$\bar{G}_{ij}^I = -\bar{G}_{ij} \cdot \bar{I}_R, \quad (2-9)$$

where  $\bar{I}_R$  is the reflection dyad  $\bar{I}_R = \mathbf{e}_x \mathbf{e}_x + \mathbf{e}_y \mathbf{e}_y - \mathbf{e}_z \mathbf{e}_z$ . The dyadic product inverts the sign of the first two columns of the matrix of the dyadic Green's function, but leaves the third column unchanged.

### Solution of the System of Equations

Now we are able to consider simultaneously all of the dipoles within the lattice. As a result, this will create a matrix equation, which we will denote as the Dipole Matrix Equation. To solve Eq. (2-1) for N dipoles simultaneously, we form the matrix equation:

$$\left( \bar{B} + \bar{A} + \bar{R} \right) \bar{P} = \bar{E}_0. \quad (2-10)$$

$\bar{B}$  is a diagonal matrix of the inverse of the dipole polarizabilities.  $\bar{A}$  includes the effects of the direct interaction between the dipoles (Eq. (2-3)).  $\bar{A}$  involves the dyadic Green's function, and thus distance vectors between each dipole.  $\bar{R}$  includes the effect of the reflection interaction between the dipoles and the

surface (Eq. 4), which includes the Sommerfeld integral terms and the image dyadic Green's function, thus also the distance vectors between the dipoles (Schmehl *et al.*, 1997).

### Method of Solution

The size of the  $3N \times 3N$  non-sparse coefficient matrix  $\bar{B} + \bar{A} + \bar{R}$  can be so large such that using a direct matrix inversion method to solve for  $\bar{P}$  is impractical. Methods which employ iterative procedures is an alternative to efficiently solve for the system of equation. During the development of DDSURF, several iterative procedures were investigated. Complex conjugate gradient-type (CCG) methods such as the biconjugate gradient (BCG) method and conjugate gradient squared (CGS) method were investigated along with several forms of the Quasi-Minimal Residual (QMR) method. Due to the characteristics of the coefficient matrix, the convergence towards an accurate answer is dependent on scattering feature size and refractive index. The effectiveness of the iterative procedure was based on the ability to converge to an accurate answer in the most reasonable amount of time. DDSURF uses the QMR method developed by Freund (1992). The QMR method follows Algorithm 3.2 in Freund (1992).

The QMR method requires approximately  $O(N^2)$  mathematical operations per iteration. When a large number of dipoles are used to model a feature, the computational time would become significantly longer. To reduce the number of operations required, a two-dimensional fast Fourier transform (FFT) is applied to the system of equations (Schmehl, 1994, Nebeker *et al.* 1996). Implementation of the FFT reduces the operations per iteration to  $O(N \log N)$ , which offers a significant amount of savings in time.

The QMR method iterates towards a solution to the Dipole Matrix Equation (Eq. 9) to a point where the Euclidean norm of the residual was smaller than the convergence criterion.

$$\bar{Z}^{(iter)} = \bar{E}_{inc} - (\bar{B} + \bar{A} + \bar{R})\bar{P}^{(iter)} \text{ (residual)} \quad (2-11)$$

$$\frac{\|\bar{Z}^{(iter)}\|_2}{\|\bar{E}_{inc}\|_2} \leq 10^{-n} \text{ (convergence criterion)} \quad (2-12)$$

$$\|(\ )\|_2 = \left( \sum_{i=1}^{3N} (\ ) \cdot (\ )^* \right)^{1/2} \text{ (Euclidean (2) norm)} \quad (2-13)$$

### II.2 Computation of the External Field

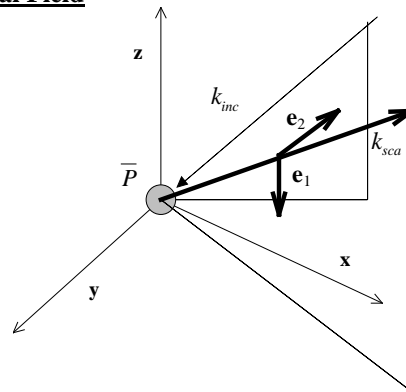


Figure 2.3. Geometrical configuration for far field scattering.

Once DDSURF has reached convergence, the user can execute FFSURF to calculate the far field electric field, the irradiance, the total scattering cross-section, and the differential scattering cross-section. From the particle to the receiving detector, there are two scattering components to consider. One component is

the field radiated directly from the dipole without interaction with a surface. Assuming that  $r \gg r_j$ , and in terms of the scattering frame of reference as shown in Fig. 2.3, this is:

$$\bar{E}_{direct,i} = \frac{k_0^2}{\epsilon_0} \sum_{j \neq i} \bar{G}_{ij} \cdot \bar{P}_j \quad (2-14a)$$

If we were to consider this in terms of the scattering reference frame, this reduces to:

$$\bar{E}_{sca,direct}(r) = k_0^2 \frac{\exp(ik_0 r)}{4\pi r} \sum_{j=1}^N \left\{ \exp(-i \mathbf{k}_{sca} \cdot \mathbf{r}_j) \left[ (\bar{P}_j \cdot \mathbf{e}_1) \mathbf{e}_1 + (\bar{P}_j \cdot \mathbf{e}_2) \mathbf{e}_2 \right] \right\} \quad (2-14b)$$

by direct scattering from each dipole.

The second component of the field that is present at the detection region is the field that is radiated from the dipole, but also reflects off of the surface. In the scattering far-field, the Fresnel scattering coefficients can be used to account for the interaction with the surface. We can use Eq. (2-14b) to determine the reflected component of the direct field. To determine the reflected field component, the method of images altered with the Fresnel coefficients to account for the dielectric surface:

$$\bar{E}_{sca,refl}(r) = k_0^2 \frac{\exp(ik_0 r)}{4\pi r} \sum_{j=1}^N \left\{ \exp(-i \mathbf{k}_{sca}^I \cdot \mathbf{r}_{I,j}) \left[ R^{TM} (\bar{P}_j \cdot \mathbf{e}_1) \mathbf{e}_1 + R^{TE} (\bar{P}_j \cdot \mathbf{e}_2) \mathbf{e}_2 \right] \right\}, \quad (2-15)$$

where  $R^{TM}$  and  $R^{TE}$  are the Fresnel reflection coefficients which are defined in Appendix B.

Summing Eqs. (2-14b) and (2-15) we get:

$$\bar{E}_{sca}(r) = k_0^2 \frac{\exp(ik_0 r)}{4\pi r} \sum_{j=1}^N \left\{ \begin{array}{l} \exp(-i \mathbf{k}_{sca} r_j) \left[ (\bar{P}_j \cdot \mathbf{e}_1) \mathbf{e}_1 + (\bar{P}_j \cdot \mathbf{e}_2) \mathbf{e}_2 \right] + \\ \exp(-i \mathbf{k}_{sca} r_j) \left[ R^{TM} (\bar{P}_j \cdot \mathbf{e}_1) \mathbf{e}_1 + R^{TE} (\bar{P}_j \cdot \mathbf{e}_2) \mathbf{e}_2 \right] \end{array} \right\}. \quad (2-16)$$

The far-field irradiance can be found by multiplying the scattered electric field by its complex conjugate:

$$I_{sca}(r) = \bar{E}_{sca}(r) \bar{E}_{sca}^\Psi(r). \quad (2-17)$$

Once the irradiance is found, the total scattering cross-section,  $C_{sca}$ , and the differential scattering cross-section,  $dC_{sc} / d\Omega$ , can be determined.

The total scattering cross section,  $C_{sca}$  is defined as (Bawolek, 1991):

$$C_{sca} = \frac{P_{sca}}{I_0} \quad (2-18)$$

where  $P_{sca}$  is the power collected by some detector, and  $I_0$  is the incident irradiance at the center of the Gaussian beam.

The differential scattering cross-section is defined as the energy scattered per unit time into a unit solid angle about a certain direction. Bohren and Huffman<sup>11</sup> derived an expression for this:

$$\frac{dC_{sca}}{d\Omega} = \lim_{\Omega \rightarrow 0} \left( \frac{C_{sca}}{\Omega} \right) \approx \frac{I_{sca} A}{I_0 (A/r^2)} = \frac{r^2 I_{sca}}{I_0}, \quad (2-19)$$

where  $I_{sca}$  is the scattered irradiance, A is the detection area scattered to, r is the distance from the particle to the observation point, and  $\Omega$  is the solid angle.

### III. COMPARISON WITH EXPERIMENT

Experiments were conducted at Arizona State University by Greg Starr to determine the validity and accuracy of DDSURF and FFSURF. Light scattering measurements from a number of feature configurations were conducted using a scatterometer and the ring/wedge detector shown in Figure 3.1.

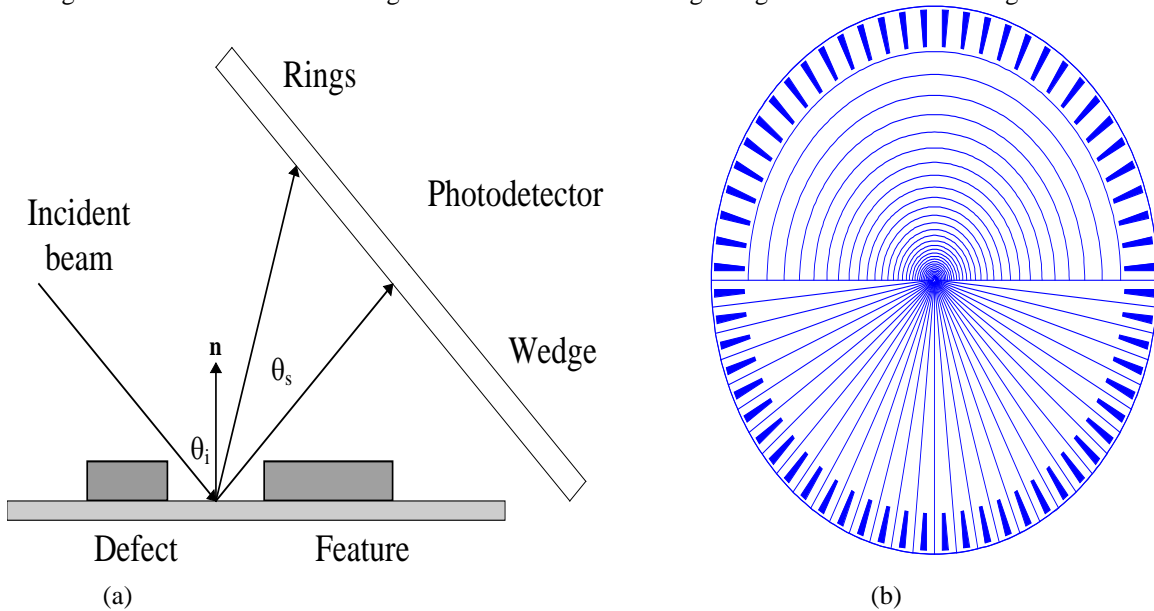


Figure 3.1 (a) Side view of beam, substrate and photodetector configuration; (b) photodetector ring/wedge configuration

The ASU scatterometer can use a helium-neon laser beam ( $\lambda = 632.8$  nm), or an argon-ion laser beam ( $\lambda = 488$  nm) to create the scattering signature from the feature on a surface. The ring/wedge photodetector which consists of 32 rings and 32 wedges that act as individual detectors detect the scattered light from the feature on the surface. The detector is centered on the specular, and the incident angle ( $\theta_i$ ) has been studied at 45 deg and 70 deg from the surface normal, and the distance from the feature to the detector center is approximately 8.2 mm.

#### III.1 Complex Shapes on Surface

DDSURF has been used to model light scattering from  $\text{SiO}_2$  patterned features present within the Arizona State University/Semiconductor Research Corporation block of the SEMATECH patterned wafer defect standard die. PSL spheres were deposited on the patterned wafer to act as particle contamination. An overhead view of the SEM photograph and the dipole configuration of the  $\text{SiO}_2$  cornered feature with a PSL sphere contaminant are shown in Figures 3.2.

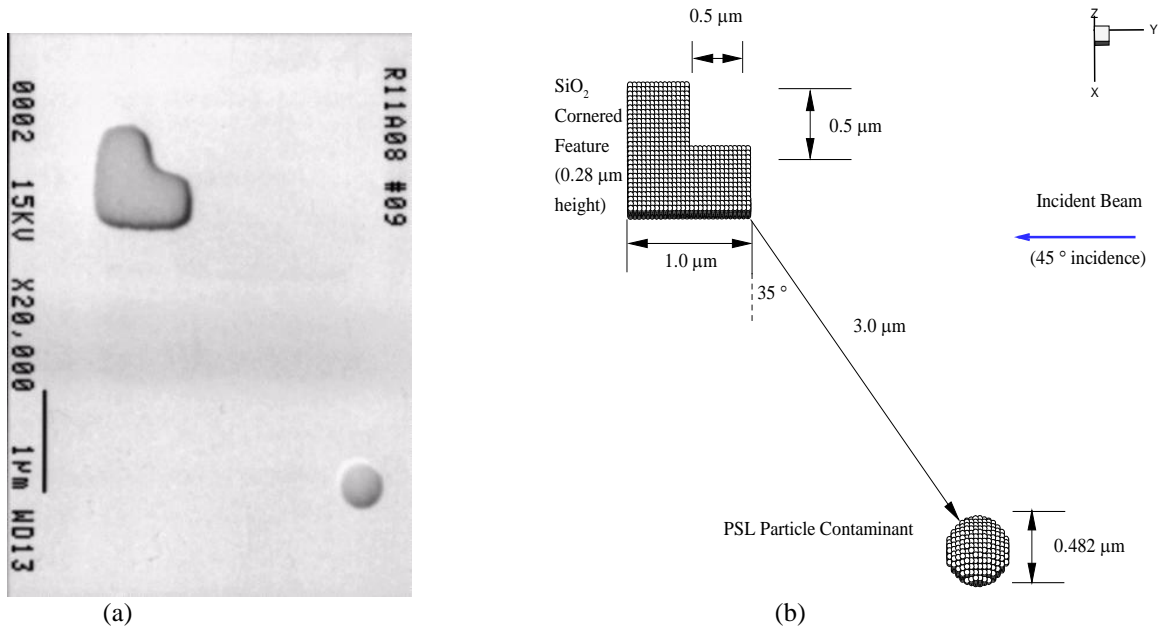


Figure 3.2 (a) SEM photo of a SiO<sub>2</sub> cornered feature with a PSL particle contaminant; (b) dipole configuration used for the DDSURF modeling of the features.

Figure 3.3a shows the computed irradiance distribution for scattering from the SiO<sub>2</sub> corner only, at the region where the detector occupied during the experiment. Figure 3.3b shows the irradiance distribution of the scattering when the particle contaminant is introduced.

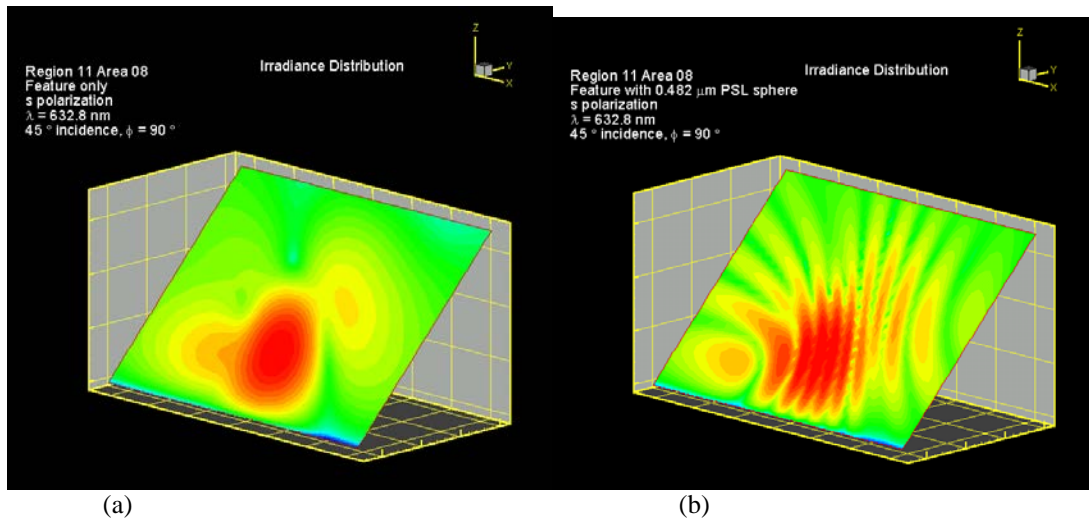
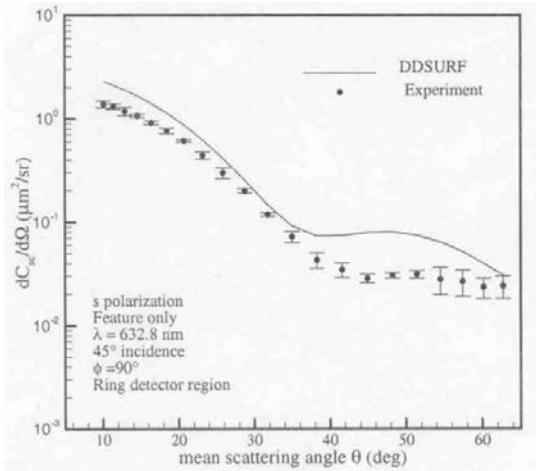
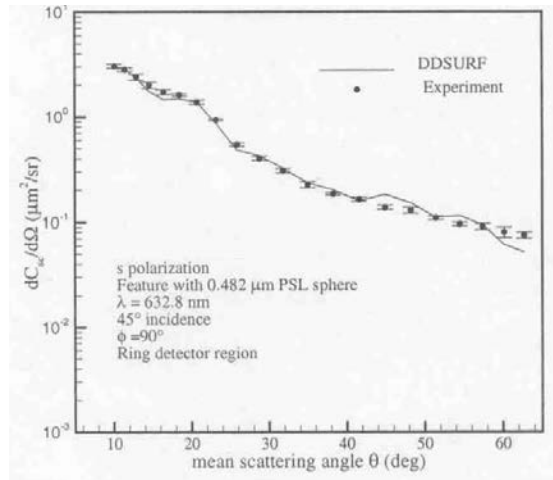


Figure 3.3 Irradiance distribution on detector plane for SiO<sub>2</sub> cornered feature: (a) feature only, (b) feature with contaminating 0.482 µm PSL sphere on a silicon surface.

Figures 3.4(a)-(b) and Figures 3.5(a)-(b) show the quantitative comparisons of the differential scattering cross-sections found by experiments conducted at Arizona State University and the results from DDSURF. We see that the comparisons are very close to each other.

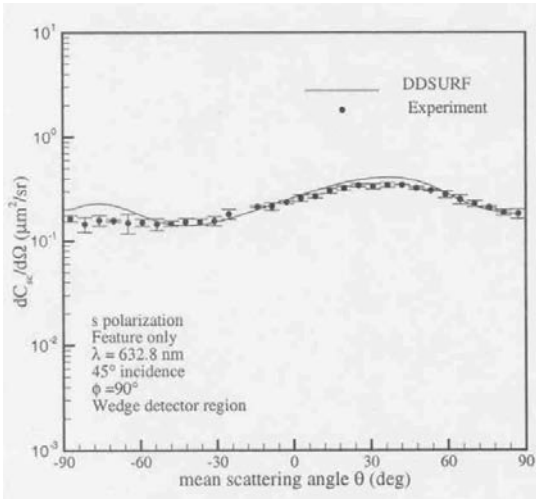


(a)

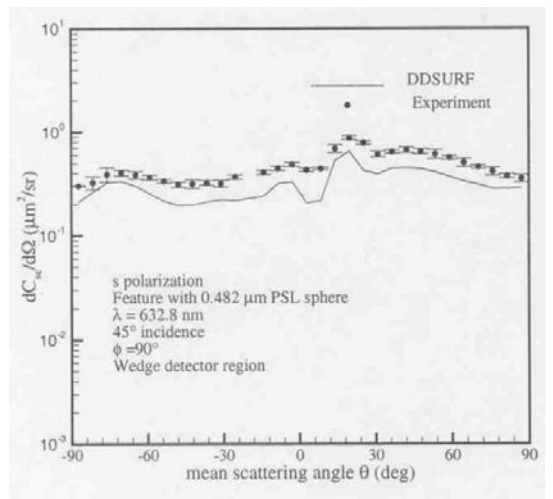


(b)

Figure 3.4 (a)  $dC_{sca}/d\Omega$  angular variation in ring region for  $\text{SiO}_2$  cornered feature: (a) feature only, (b) feature with  $0.482 \mu\text{m}$  PSL contaminant sphere on a silicon surface.



(a)



(b)

Figure 3.5 (a)  $dC_{sca}/d\Omega$  angular variation in wedge region for  $\text{SiO}_2$  cornered feature: (a) feature only, (b) feature with  $0.482 \mu\text{m}$  PSL contaminant sphere on a silicon surface.

### III.2 Features on Filmed Surfaces

To study the accuracy of the filmed-based DDSURF, which is also referred to as DDFILM, experiments were conducted at Arizona State University for spherical particles which were placed on a filmed substrate. The scattered field was measured using the laser scatterometer at the Laser Diagnostics laboratory. Figures 3.6(a) and 3.6(b) show the p and s polarization cases respectively for scattering from a  $0.305 \mu\text{m}$  PSL sphere which lies upon a silicon substrate with a  $0.25 \mu\text{m}$   $\text{SiO}_2$  film. The detection region used was that for the ring section of the ring/wedge detection system shown in Fig. 3.1. The incident beam was at 70 degrees from the surface normal. The comparisons are good and the DDFILM results lie within the uncertainty levels of the experiment.

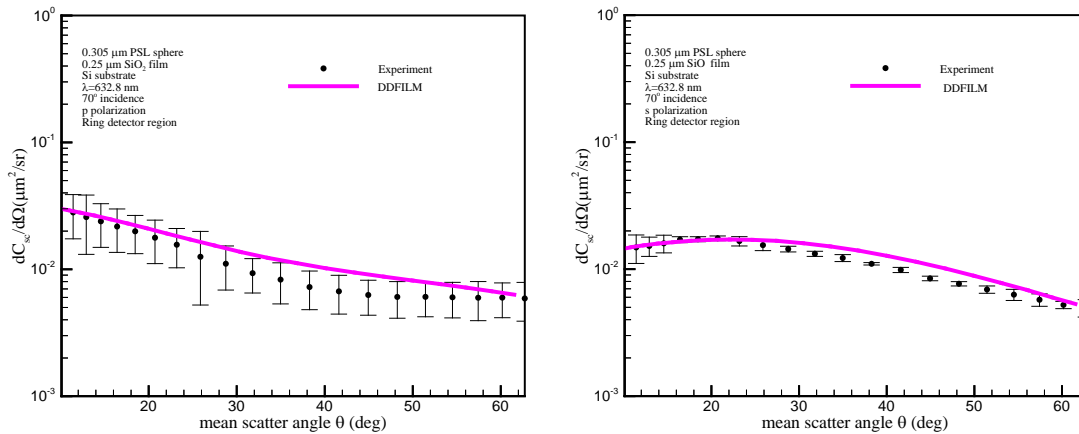


Figure 3.6  $dC_{sca}/d\Omega$  angular variation in ring region for a  $0.305 \mu\text{m}$  PSL sphere on a Si substrate with a  $0.25 \mu\text{m}$   $\text{SiO}_2$  film. Incident beam is at  $70^\circ$  incidence,  $632.8 \text{ nm}$  wavelength, and: (a) p polarization, (b) s polarization.

Figure 3.7 shows comparisons of the differential scattering cross-section found using the Filmed DDSURF (DDFILM) and experiment conducted at Arizona State University using a point silicon detector. The scattering angle is the theta distribution along the plane of incidence. A  $.305 \text{ mm}$  PSL is place upon a Si substrate which now has a  $\text{SiO}_2$  film which is  $0.7849 \text{ mm}$  thick. We see from the figure that DDFILM follows the scattering measurements very well. No uncertainty bars were included in the data, as the experiment was only conducted once.

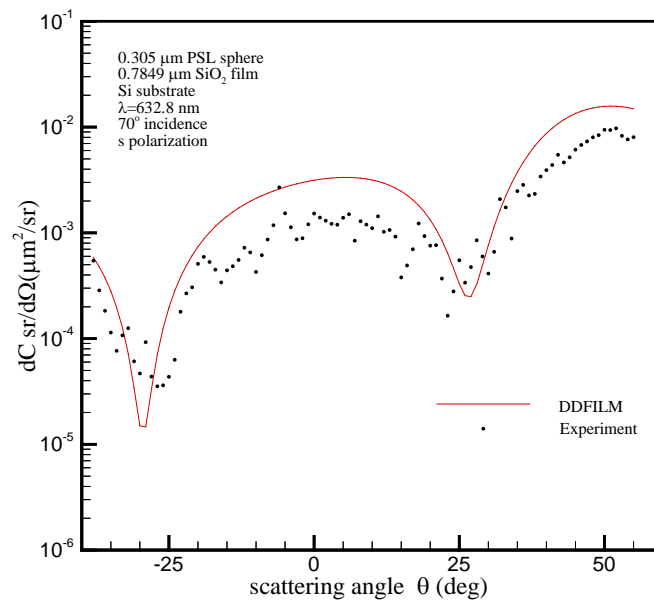


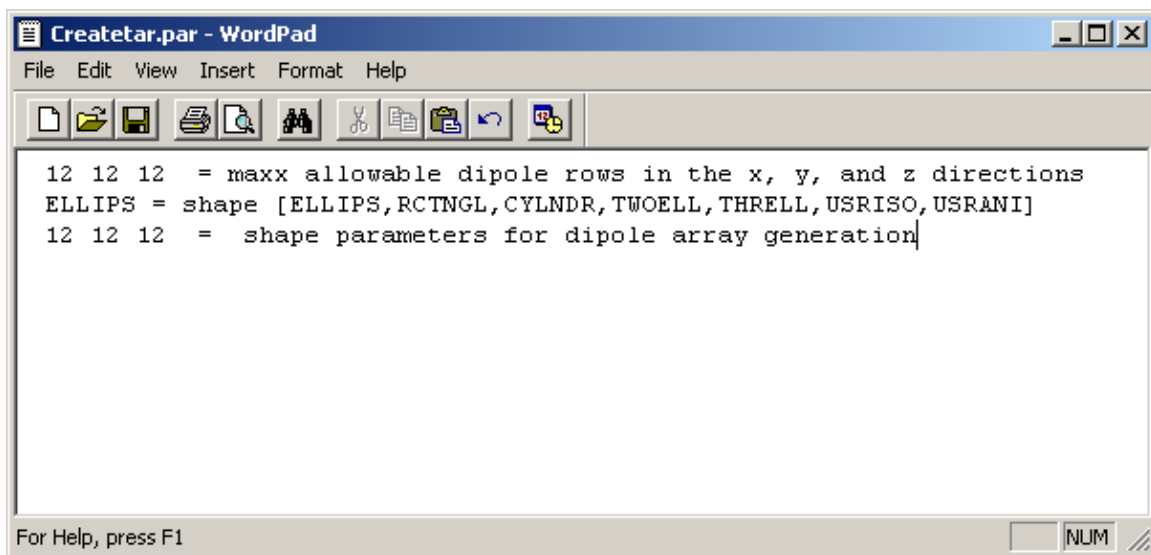
Figure 3.7  $dC_{sca}/d\Omega$  angular variation in the plane of detection for a  $0.305 \mu\text{m}$  PSL sphere on a Si substrate with a  $0.25 \mu\text{m}$   $\text{SiO}_2$  film. Incident beam is at  $70^\circ$  incidence,  $632.8 \text{ nm}$  wavelength, and s polarization.

## IV. DDSURF 2007 SOFTWARE

In this section, we describe each component of the DDSURF 2007 code, starting first with the dipole arrangement component called CREATETAR, moving to the component where the internal field of the scatter is computed called DDSURF, and finally to the component where the far-field scattering is computed called FFSURF.

### IV.1 CREATETAR

**CREATETAR** is a code that is used to create the feature shape input file, *shape.dat*, to be used to describe the dipole positions for DDSURF which is the code used to predict electromagnetic scattering characteristics from small features that exist either in free space or on flat surfaces. CREATETAR requires an input parameter file, which is called *createtar.par*. The following is an example of *createtar.par*.



```
12 12 12 = maxx allowable dipole rows in the x, y, and z directions
ELLIPS = shape [ELLIPS,RCTNGL,CYLNDR,TWOELL,THRELL,USRISO,USRANI]
12 12 12 = shape parameters for dipole array generation
```

#### IV.1.1 CREATETAR Input Parameters

##### **12 12 12 = max allowable dipole rows in the x, y, and z directions**

This input line is the specification of the dynamically allocated memory size for the storage of the number of dipole rows for the feature in the  $x$ ,  $y$ , and  $z$  directions. If the number of rows is exceeded by the modeled feature, an error will occur. This designates the size of the computer swap file, *c:\windows\emm386.swp* for Win95. If there is not enough memory available in the computer, an error will occur, and the user will have to either free up memory in the computer, or decrease the number of dipole rows that will be used. (These parameters must be integers)

##### **ELLIPS = shape [ELLIPS,RCTNGL,CYLNDR,TWOELL,THRELL,USRISO,USRANI]**

This input line is the specification of the feature shape that is of interest. The shapes that are specified are ellipsoid or spherical features (**ELLIPS**), rectangular block features (**RCTNGL**), cylindrical features (**CYLNDR**), two touching ellipsoid features of the same or different materials aligned along the  $x$  axis (**TWOELL**), three ellipsoid features touching in a series of the same or different materials aligned along the  $x$  axis (**THRELL**), the user specified shape for isotropic materials (**USRISO**), and user specified shape



for anisotropic materials (USRANI). To specify a feature shape not given as an option, one needs to create an input file named *shapein.dat*. This data file will include the number of dipoles used, the x, y, and z positions of the dipoles, as well as the type of material each dipole represents. An example for *shapein.dat* is given in Section 4.2.3.

### 12 12 12 = shape parameters for dipole array generation

This input line is the x1, x2, and x3 dipole shape parameters for the input feature. This line will be ignored if an user-defined shape is specified by *shapein.dat*. For the pre-defined shapes available x1, x2, and x3 may designate different parameters.. The meanings are the following for the given shapes:

#### ELLIPS

x1: max number dipoles in x direction  
 x2: max number dipoles in y direction  
 x3: max number dipoles in z direction

#### RCTNGL

x1: number of dipoles in x row direction  
 x2: number of dipoles in y row direction  
 x3: number of dipoles in z row direction

#### CYLNDR

x1: number dipoles in long axis direction  
 x2: number dipoles in diameter of cylinder axis direction  
 x3: direction of cylindrical axis (1 = x, 2 = y, 3 = z)

#### TWOELL

x1: number of dipoles in x row direction (each ellipse)  
 x2: number of dipoles in y row direction (each ellipse)  
 x3: number of dipoles in z row direction (each ellipse)

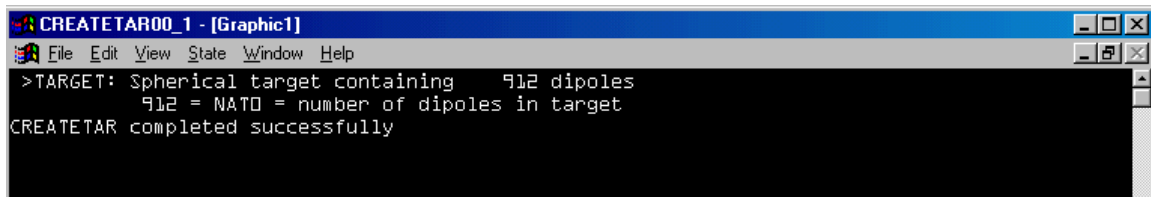
#### THRELL

x1: number of dipoles in x row direction (each ellipse)  
 x2: number of dipoles in y row direction (each ellipse)  
 x3: number of dipoles in z row direction (each ellipse)

For example, for this particular case, a rectangular feature has been specified, with 12 dipoles rows in the x direction at the center, 12 dipoles in the y direction at the center, and 12 dipoles in the z direction at the center.

### IV.1.2 Execution of CREATETAR

Once these parameters have been set, the user can double click the *createtar07.exe* file, which uses *createtar.par*. After execution, the dipole array data file called *shape.dat* is created to be used by DDSURF to model the scattering feature. The screen output looks as the following:



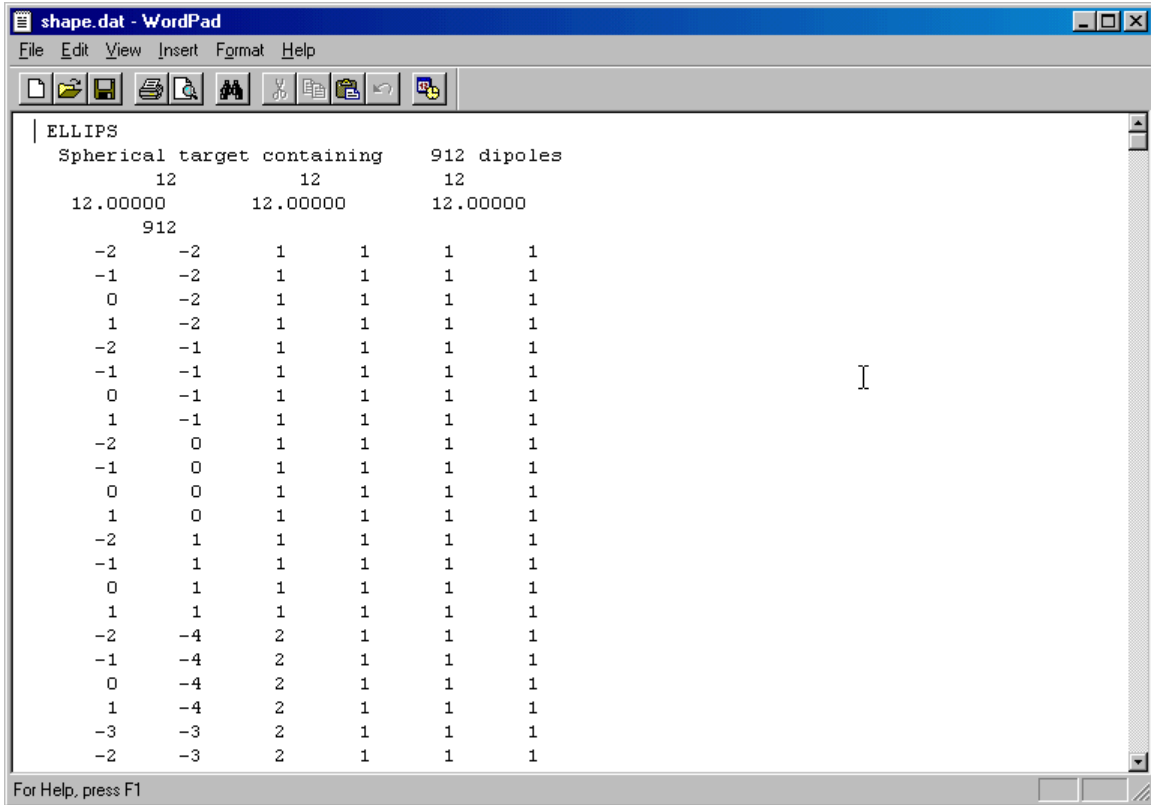
```

CREATETAR00_1 - [Graphic1]
File Edit View State Window Help
>TARGET: Spherical target containing 912 dipoles
          912 = NATO = number of dipoles in target
CREATETAR completed successfully
  
```

The output describes the feature shape which has been created and the total number of dipoles that were used in the lattice array. The next section will describe the *shape.dat* file that is created by CREATETAR.

### IV.1.3 SHAPE.DAT

Once CREATETAR has been executed, the output file, *shape.dat*, is created. This will be used by DDSURF as the dipole lattice array designation. The following shows a portion of a *shape.dat* file that was created by CREATETAR for an ellipsoidal feature that is 12 x 12 x 12 dipoles in dimension:



```
ELLIPS
Spherical target containing      912 dipoles
      12      12      12
12.00000 12.00000 12.00000
      912
      -2      -2      1      1      1      1
      -1      -2      1      1      1      1
      0      -2      1      1      1      1
      1      -2      1      1      1      1
      -2      -1      1      1      1      1
      -1      -1      1      1      1      1
      0      -1      1      1      1      1
      1      -1      1      1      1      1
      -2      0      1      1      1      1
      -1      0      1      1      1      1
      0      0      1      1      1      1
      1      0      1      1      1      1
      -2      1      1      1      1      1
      -1      1      1      1      1      1
      0      1      1      1      1      1
      1      1      1      1      1      1
      -2      -4      2      1      1      1
      -1      -4      2      1      1      1
      0      -4      2      1      1      1
      1      -4      2      1      1      1
      -3      -3      2      1      1      1
      -2      -3      2      1      1      1
```

The first lines describe the feature, the maximum spacings available, and the spacings used, as well as the number of dipoles used to fill the lattice array. Next is the description of the lattice positions where dipoles occupy the array. The first three columns describe the x, y, and z positions respectively. The next three columns describe the materials used in the feature, which the quantities in the x, y, and z directions. For the example shown, the feature contains a material #1, to be designated in DDSURF, and the feature is isotropic and uniform, no other material properties are needed. If the user desires to add more feature materials, i. e., a material #2, the user must change the values manually by the user.

### IV.1.4 Example Feature Shape Designations (Pre-defined Shapes)

In this section, we describe the steps necessary to develop the feature shapes of interest to the user if the shapes are included in the various available pre-defined shapes.

#### IV.1.4.1 Ellipse Features

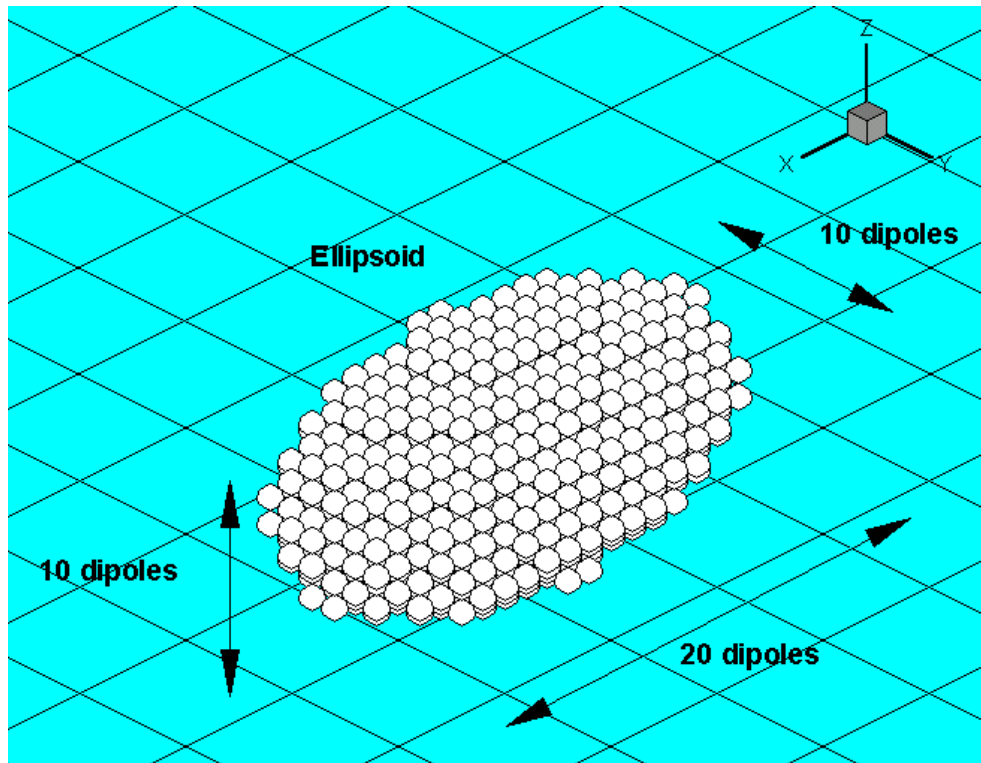
The following is an example when ELLIPS is designated as 20 10 10, which means an ellipsoid shaped feature has a maximum of 20 dipole rows in the x direction, a maximum of 10 dipole rows in the y direction, and a maximum of 10 dipole rows in the z direction. Out of the 2000 possible lattice sites, 1032 sites are filled with dipoles to best fit an ellipsoid.

```

Createtar.par - WordPad
File Edit View Insert Format Help
[Icons]
20 10 10 = maxx allowable dipole rows in the x, y, and z directions
ELLIPS = shape [ELLIPS,RCTNGL,CYLNDR,TWOELL,THRELL,USRISO,USRANI]
20 10 10 = shape parameters for dipole array generation
For Help, press F1
NUM

```

Note that the first line, which is the memory allocation line, must be equal to or greater than the total number of spacings of the actual feature. Once these parameters have been set, the user executes *createtar07.exe*. The following shows a representation of the dipole arrangement which was created using the dipole array inputs:



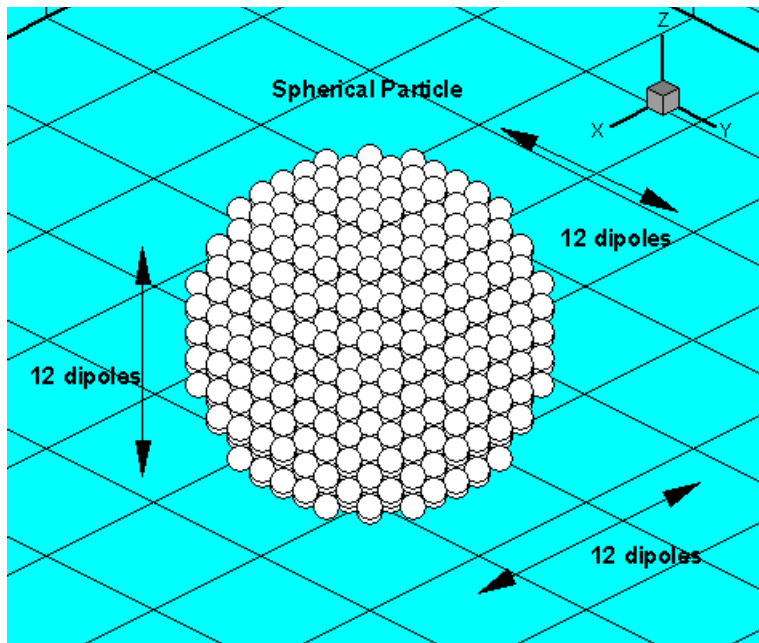
CREATETAR fills the lattice with dipoles to fit within 20 dipole spacings in the x direction, and 10 dipole spacings in the y and z directions.

#### IV.1.4.2 Spherical Features

The ELLIPS shape indicator can create ellipsoid shaped features, including spheres. When the shape indicators of the third line of *createtar.par* are designated to be the same values, a spherical feature can be created. Consider creating a spherical feature with a maximum of 12 rows of dipoles in any direction, the important input lines of *createtar.par* would have the form:

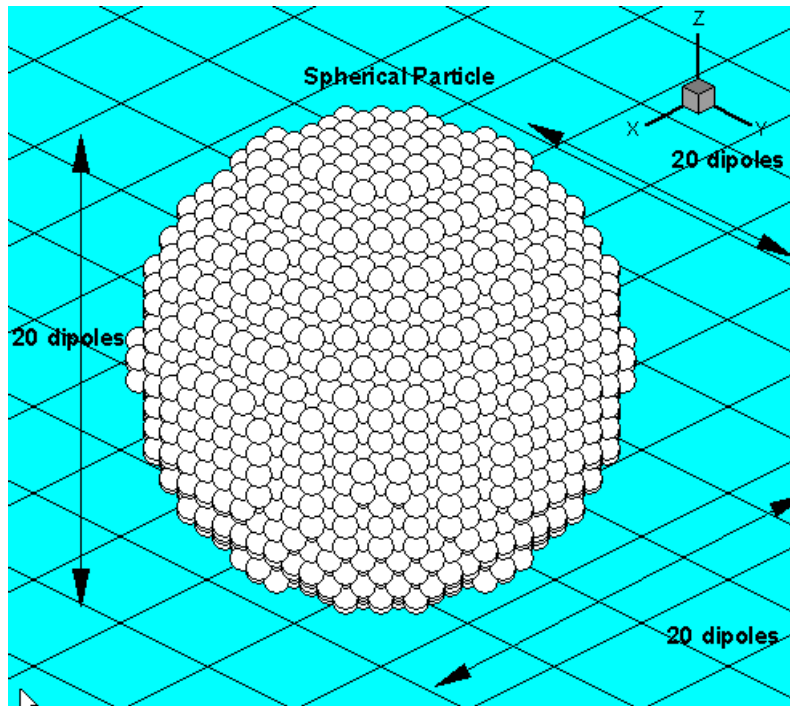
```
12 12 12 = maxx allowable dipole rows in the x, y, and z directions
ELLIPS = shape [ELLIPS,RCTNGL,CYLNDR,TWOELL,THRELL,USRISO,USRANI]
12 12 12 = shape parameters for dipole array generation
```

For Help, press F1



CREATETAR fits the dipoles into the nearest approximation to a sphere as possible within the limits indicated. Out of a possible 1728 dipole sites created by the 12 x 12 x 12 lattice structure, 912 sites are filled with dipoles.

Since dipoles are discrete points, a spherical feature would be more accurately modeled as the number of dipoles used is increased. If a 20 x 20 x 20 dipole row configuration was indicated in *createtar.par*, out of the possible 8000 lattice sites, 4224 sites are filled with dipoles.



Increasing the number of dipoles has improved the modeling of the shape, but at the same time, it increases the computational time needed, as well as memory usage for the computer. The user must determine how to best apply the balance between computational considerations and modeling considerations.

#### IV.1.4.3 Cylindrical Features

When the shape designated is CYLNDR, the three inputs are:

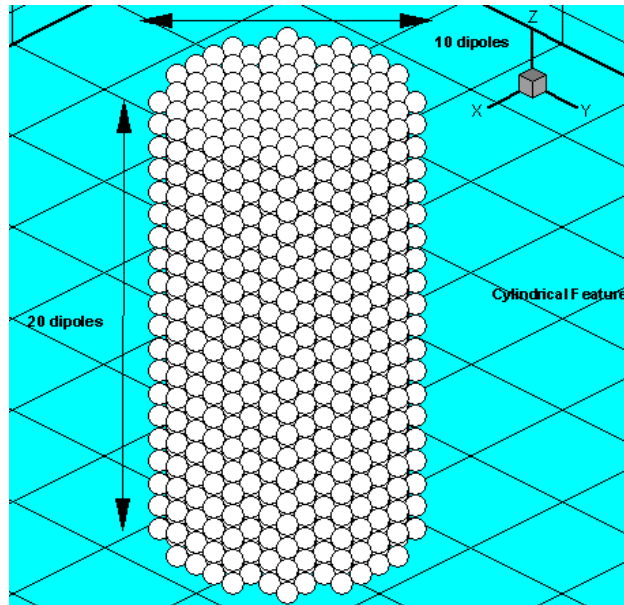
(cylinder length, cylinder diameter, Axis of Symmetry)

To designate the Axis of Symmetry **1** indicates x direction; **2** indicates y direction; **3** indicates x direction. The following shows three examples of a cylinder with 20 dipole rows in length, and 10 dipole rows in diameter, with the three axis of symmetry:

```

Createtar.par - WordPad
File Edit View Insert Format Help
10 10 20 = maxx allowable dipole rows in the x, y, and z directions
CYLNDR = shape [ELLIPS,RCTNGL,CYLNDR,TWOELL,THRELL,USRISO,USRANI]
20 10 3 = shape parameters for dipole array generation
For Help, press F1
NUM

```

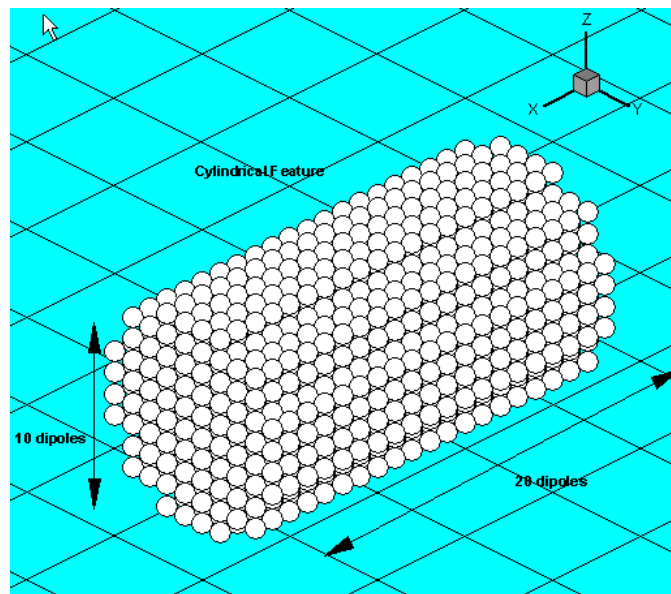


The user can also specify cylinders in which the axis of symmetry is also in the x or y directions. Examples of these cases are as follows:

```

Createtar.par - WordPad
File Edit View Insert Format Help
[Icons]
20 10 10 = maxx allowable dipole rows in the x, y, and z directions
CYLNDR = shape [ELLIPS,RCTNGL,CYLNDR,TWOELL,THRELL,USRISO,USRANI]
20 10 1 = shape parameters for dipole array generation
For Help, press F1
NUM

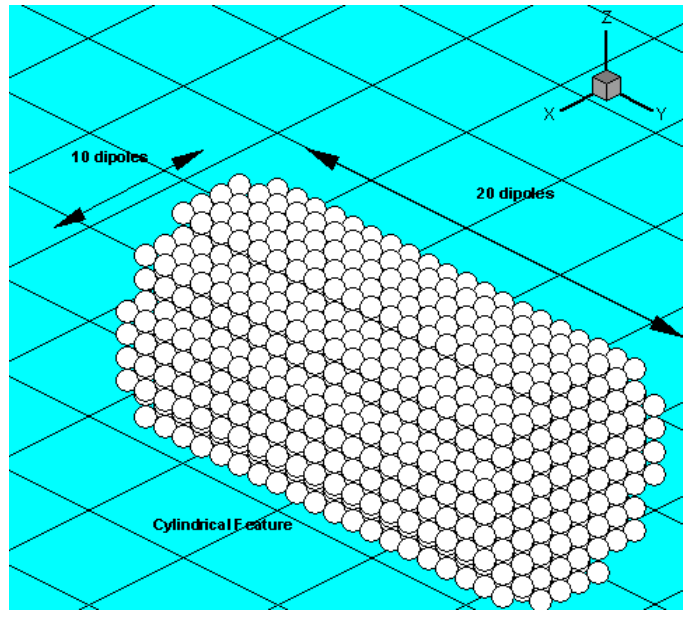
```



```

Createtar.par - WordPad
File Edit View Insert Format Help
10 20 10 = maxx allowable dipole rows in the x, y, and z directions
CYLNDR = shape [ELLIPS,RCTNGL,CYLNDR,TWOELL,THRELL,USRISO,USRANI]
20 10 2 = shape parameters for dipole array generation
For Help, press F1
NUM

```



It is important to note the modifications necessary to the MXNX, MXNY, and MXNZ line. When the cylinder is along the x axis, the max number of x spacings must be changed to 20, while if the cylinder is along the y axis, the max number of y spacings must be changed to 20. If the number of dipole spacings in the feature exceeds the maximum specified, a user-error will occur when DDSURF is executed.

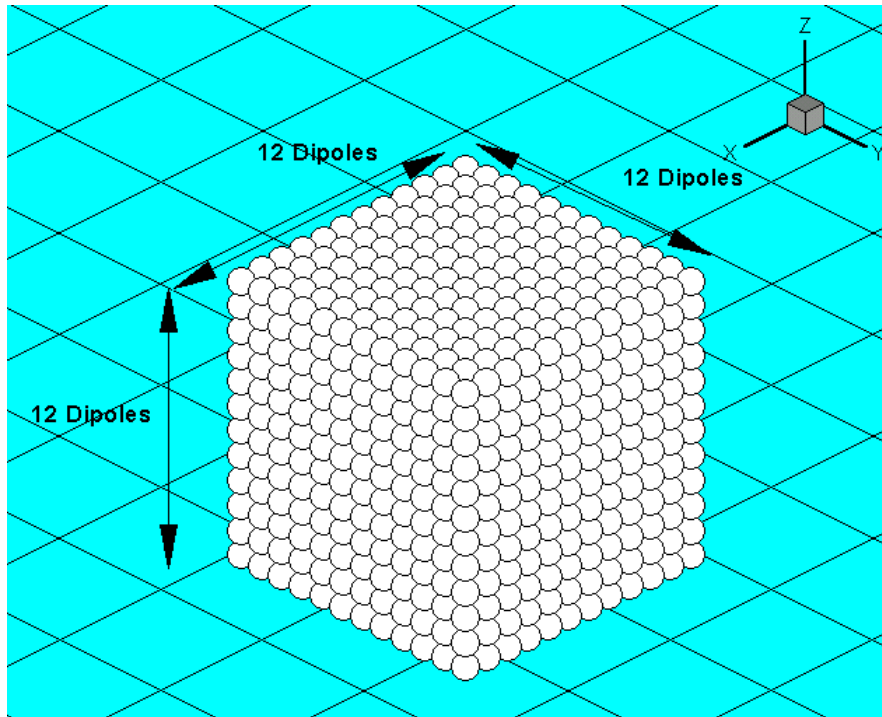
IV.1.4.4 Rectangular Features

Designation of a rectangular prism shape is the simplest shape created by CREATETAR. The user specifies the number dipole spacings in the x, y, and z directions, and an array of those dimensions are filled to make the rectangular shape. The following shows an example of a 12 x 12 x 12 dipole row rectangular feature:

```

Createtar.par - WordPad
File Edit View Insert Format Help
12 12 12 = maxx allowable dipole rows in the x, y, and z directions
RCTNGL = shape [ELLIPS,RCTNGL,CYLNDR,TWOELL,THRELL,USRISO,USRANI]
12 12 12 = shape parameters for dipole array generation
For Help, press F1
NUM

```



#### IV.1.4.5 Two Ellipse, Three Ellipse Shape

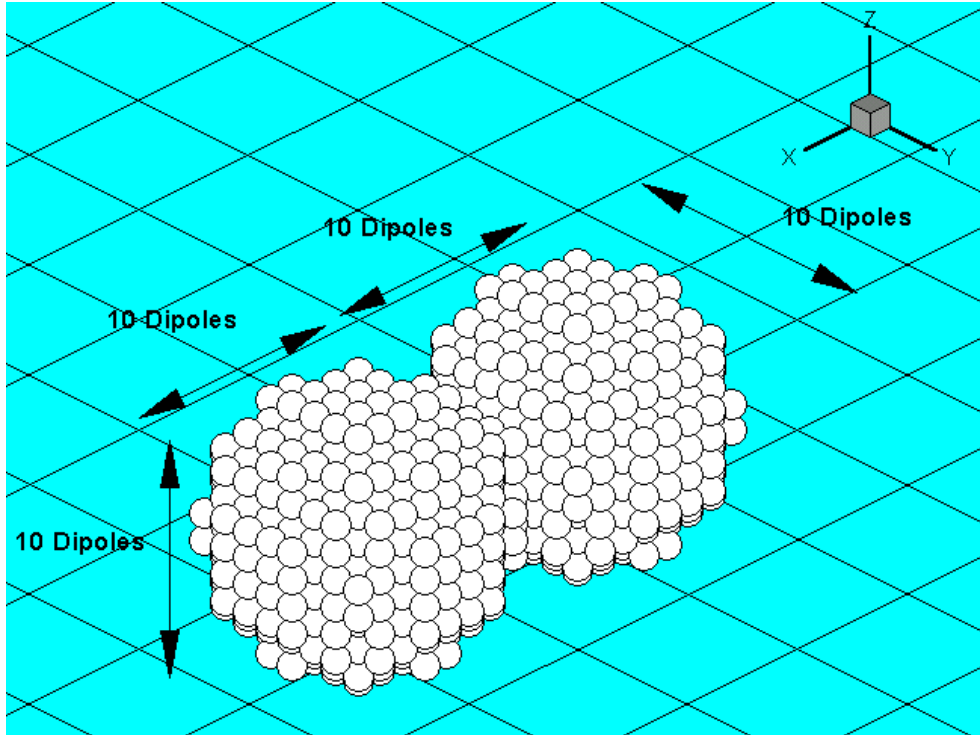
Two other options, the “Two Ellipse” and the “Three Ellipse” allow the user to create 2 or 3 ellipses in a row along the x directions. With reference to the semiconductor industry, these would be similar to when two or three spherical particles are attached to each other. NX, NY, and NZ are designated for each feature, and they will be the same, for example the following creates two ellipses along the x axis:

```

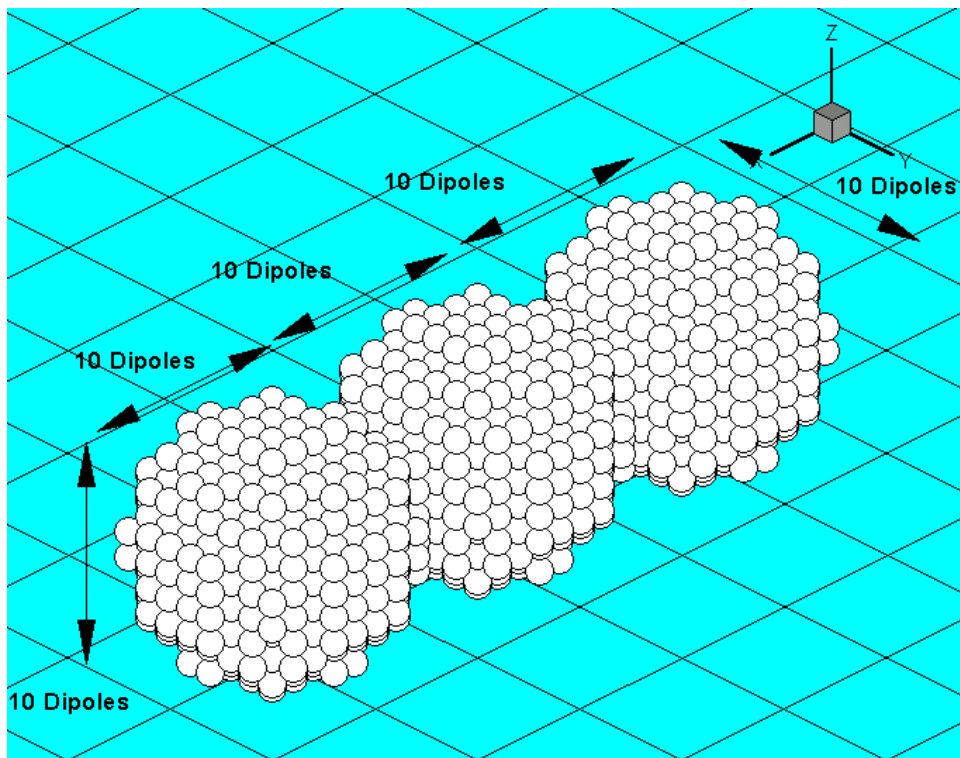
Createtar.par - WordPad
File Edit View Insert Format Help
[Icons: Undo, Redo, Save, Print, Find, Zoom, Copy, Paste, Undo, Help]
20 10 10 = maxx allowable dipole rows in the x, y, and z directions
TWOELL = shape [ELLIPS,RCTNGL,CYLNDR,TWOELL,THRELL,USRISO,USRANI]
10 10 10 = shape parameters for dipole array generation
For Help, press F1
NUM

```

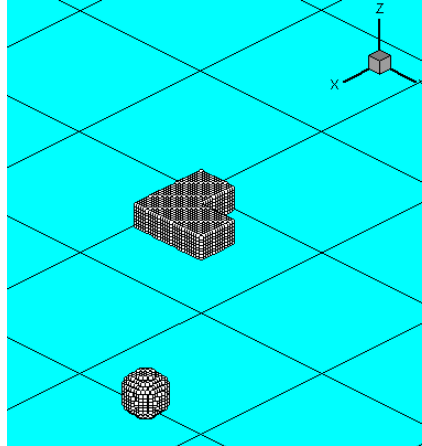




An example of the three ellipse case is as follows:



#### IV.1.4 Example Feature Shape Designations (User Defined Shapes)



Many shapes that will be of interest to the user will not be found in the pre-defined shapes that have been mentioned previously. One example is shown in the figures above which show a photograph of a cornered geometrical shape near a spherical contaminating particle. In order to model this, the user needs to manually create an input file which shows the dipole distribution. Next to the photograph of the feature/contaminant system is the dipole arrangement to model the feature/contaminant configuration. To apply this to DDSURF, the user will need to choose either the User Defined Isotropic (USRISO) option in which the material in the feature is isotropic (the refractive index does not change with direction) or the User Defined Anisotropic (USRANI) in which the material in the feature can be anisotropic (refractive index vary with direction). The user defined file describing the dipole arrangement needs to be called *shapein.dat*. This file will be used as an input file when executing CREATETAR. In this problem, the materials are isotropic, therefore, the *shapein.dat* file will have the following form:

```
6176
      1      1      1      1      1
      2      2      1      1      1
      3      3      1      1      1
      4      4      1      1      1
      5      5      1      1      1
      6      6      1      1      1
      7      7      1      1      1
      8      8      1      1      1
      9      9      1      1      1
     10     10      1      1      1
     11     11      1      1      1
     12     12      1      1      1
     13     13      1      1      1
     14     14      1      1      1
     15     15      1      1      1
     16     16      1      1      1
     17     17      1      1      1
     18     18      1      1      1
```

The first line of *shapein.dat* designates the total number of dipoles in the system, which is 6167 for this example. The first column is the dipole number associated with the actual dipole used at that point in the array. Next, the second, third, and fourth columns describe the dipole position in x, y, and z dipole coordinates. These positions must be integers, while the z column does not include the height above the

substrate if feature is suspended above the surface of the substrate. The fifth column describes the material, for this example, the cornered feature is of one material (#1) and the particle is of another material (#2). If we were to see the whole file, the fifth column would have the number 2 for the particle material. The material properties will be designated in DDSURF.

If the USRANI option was chosen, then each dipole position would be repeat 3 times to account for the different materials. The following shows an of *shapein.dat* for an anisotropic material with three different components. The first line accounts for x direction, the second line for the y direction material, and the third line for the z direction material. The following is an example of *shapein.dat* when "User Defined Anisotropic" is designated:

```

6176
1 1 1 1 1
1 1 1 1 2
1 1 1 1 3
2 2 1 1 1
2 2 1 1 2
2 2 1 1 3
3 3 1 1 1
3 3 1 1 2
3 3 1 1 3
4 4 1 1 1
4 4 1 1 2
4 4 1 1 3
5 5 1 1 1
5 5 1 1 2
5 5 1 1 3
6 6 1 1 1
6 6 1 1 2
6 6 1 1 3
7 7 1 1 1
7 7 1 1 2
7 7 1 1 3
8 8 1 1 1
8 8 1 1 2
8 8 1 1 3
9 9 1 1 1
9 9 1 1 2

```

When the USRISO or USRANI are chosen, the user will only need to input the maximum number of spacings allowed in the axial directions. The following was used for the cornered feature/contaminant configuration:

```

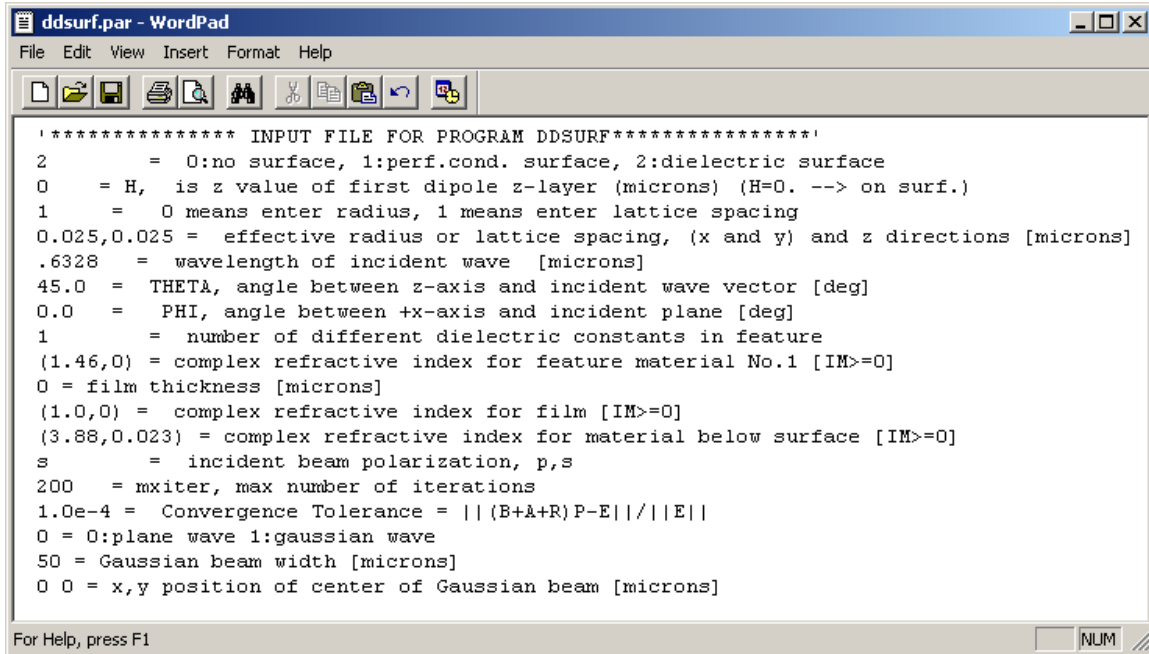
115 90 15 = maxx allowable dipole rows in the x, y, and z directions
USRISO = shape [ ELLIPS, RCTNGL, CYLNDR, TWOELL, THRELL, USRISO, USRANI ]

```

Thus, for the example case considered, the farthest dipoles in terms of x spacing were 115 dipole spacings in terms of y 90 dipole spacings, and in terms of z 15 dipole spacings.

## IV.2 DDSURF

DDSURF is the portion of the code which takes the results of CREATETAR (*shape.dat* – the feature shape input file), reads the user inputs for feature, substrate, and incident beam characteristics, and then computes the internal field of the scatterer, which will be used by FFSURF to compute the external far-field scattering. The following is the screenshot of the parameter.



```
'***** INPUT FILE FOR PROGRAM DDSURF*****'  
2      = 0:no surface, 1:perf.cond. surface, 2:dielectric surface  
0      = H, is z value of first dipole z-layer (microns) (H=0. --> on surf.)  
1      = 0 means enter radius, 1 means enter lattice spacing  
0.025,0.025 = effective radius or lattice spacing, (x and y) and z directions [microns]  
.6328  = wavelength of incident wave [microns]  
45.0   = THETA, angle between z-axis and incident wave vector [deg]  
0.0    = PHI, angle between +x-axis and incident plane [deg]  
1      = number of different dielectric constants in feature  
(1.46,0) = complex refractive index for feature material No.1 [IM>=0]  
0      = film thickness [microns]  
(1.0,0) = complex refractive index for film [IM>=0]  
(3.88,0.023) = complex refractive index for material below surface [IM>=0]  
s      = incident beam polarization, p,s  
200    = mxiter, max number of iterations  
1.0e-4 = Convergence Tolerance = ||(B+A+R)P-E||/||E||  
0      = 0:plane wave 1:gaussian wave  
50     = Gaussian beam width [microns]  
0 0    = x,y position of center of Gaussian beam [microns]
```

In the following sections, we will describe the input parameters need to describe the scattering system for use by DDSURF.

### IV.2.1 DDSURF – Input Parameters

**2 = 0:no surface, 1:perf.cond. surface, 2:real surface**

Selection of this option determines the surface characteristics in which the features lie. Setting this option to **0** will specify a “No surface” case, thus the feature is modeled to exist isolated from any surface, or in “free space”. When **1** is specified, the surface is modeled as a perfect conductor, or perfect reflecting medium, the Fresnel reflection coefficients are equal to 1. This method is also known as the *method of images*. When this is set to **2**, the surface is modeled as a real dielectric, and thus uses the Sommerfeld Integral terms to calculate the dipole interaction with the surface

**0.0 = H, is z value of first dipole z-layer [microns] (H=0. on surf.)**

When the scattering features lies on top of the substrate or the film which may be present, the feature height is specified as 0. For cases when the feature is suspended above the surface, the user can specify this distance in microns for this input.

**1 = 0 means enter radius, 1 means enter lattice spacing**

Selection of this option will determine the manner in which the feature size is implemented. When it is set to **1** (lattice Spacing” is chosen), the spacing between dipoles will be specified with the following line as follows:

```

'***** INPUT FILE FOR PROGRAM DDSURF*****'
2      = 0:no surface, 1:perf.cond. surface, 2:dielectric surface
0      = H, is z value of first dipole z-layer (microns) (H=0. --> on surf.)
1      = 0 means enter radius, 1 means enter lattice spacing
0.025,0.025 = x,y lattice spacing and z lattice spacing / effective radius [microns]

```

Lattice spacing is defined as the distance between the centers of the dipoles. For this case, the x and y spacing has been specified as .025 microns in the first input and the z spacing has been specified as 0.025 in the second spacing. Through investigation, it is recommended that the dipole spacing to be at least 1/10 to 1/20 of the incident beam wavelength to properly model the feature internal field. The ability to have different spacing between the x,y spacing and the z spacing allows for larger features to be modeled. Investigation has shown, though that when the ratio of the x,y lattice spacing to the z lattice spacing is 2.0, the comparisons tend to deviate from the original, uniform dipole spacing configuration. Examples of this are shown in Appendix B.

When 0 is specified (Effective Radius option is chosen), the effective radius of the scattering feature will be used to determine the size of the feature and the dipole spacing. Input of the radius will be in the following line, which now has the form:

```

'***** INPUT FILE FOR PROGRAM DDSURF*****'
2      = 0:no surface, 1:perf.cond. surface, 2:dielectric surface
0      = H, is z value of first dipole z-layer (microns) (H=0. --> on surf.)
0      = 0 means enter radius, 1 means enter lattice spacing
0.01   = x,y lattice spacing and z lattice spacing / effective radius [microns]

```

The effective radius specified will create a feature of volume within a sphere which has the equivalent volume as the feature being modeled.

For example, if a cubical feature is specified, and the dipole shape parameters were specified as 8 x 8 x 8. If an equivalent radius of .5 μm is specified, the actual dimensions of the rectangular feature modeled can be calculated as follows:

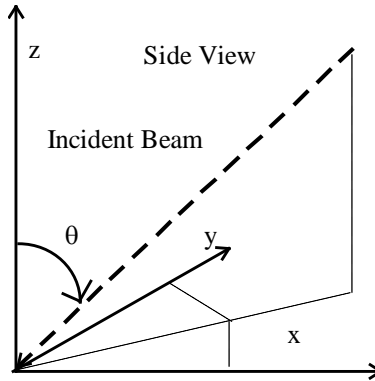
$$V = \frac{4}{3}\pi(.5)^3 = l^3 \text{ thus } l = 0.805 \mu\text{m}$$

**0.6328 = WAVE wavelength of incident wave [microns]**

This input line is the specification of the incident beam wavelength in microns. Common values are .6328 μm for the red line of the Helium-Neon laser and .488 μm for an Argon-ion laser.

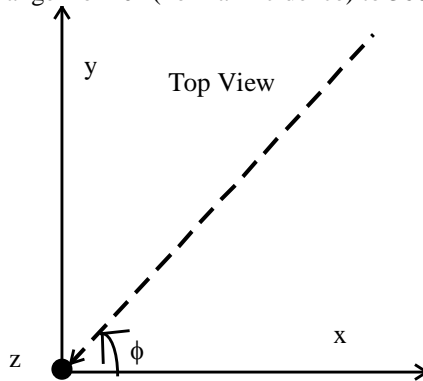
**45.0 = THETA, angle between z-axis and incident wave vector [deg]**

This input line is the specification of the angle of incidence between the z-axis (normal to the surface) and the incident beam in degrees. This value can range from 0° (normal incidence) to 90° when a surface is present, and 0° to 180° when no surface is present.



**.0 = PHI angle between +x-axis and incident plane [deg]**

This input line is the specification of the angle of incidence between the positive x axis and the incident beam in degrees. This value can range from 0° (normal incidence) to 360°.



**1 = number of different refractive indices in feature**

This input line is the specification of the number of different materials present in the features that are modeled. If 1 is specified, only one refractive index needs to be specified for the feature, if more than one is specified, the same number of refractive indices needs to be specified in the following input line. To specify the position of the various materials of a feature, a user-specified *shapein.dat* file needs to be created when considering CREATETAR. Once this is created, the user needs to add a fifth column to the data file. The number 1 will specify that the dipole will have refractive index 1, a number 2 will specify that the dipole will have refractive index 2, and so on. Please refer to Section IV.1.4 for an example of the *shapein.dat* file.

**(1.46,0.0) = complex refractive index for feature material No.1 [IM>=0]**

This is the specification of the refractive index of the material for the scattering feature. If more than one material is specified, it is possible to put multiple line inputs here for the various feature refractive indices. There is a limit of 10 feature refractive indices hard coded into DDSURF at this time. Due to the conventions used by DDSURF, the imaginary part of the refractive index should be positive.

**0 = film thickness [microns]**

This input line specifies the thickness, in microns, of the single film which lies above the infinite substrate. 0 is specified when there is no film present upon the substrate.

**(1.46,0.0) = complex refractive index for film [IM>=0]**

This input line specifies the refractive index of the film above the substrate. DDSURF conventions specify that the imaginary part should be positive.

**(3.88,0.023) = complex refractive index for material below surface [IM>=0]**

This is the specification of the refractive index of the material which acts as the surface. Due to the conventions used by DDSURF, the imaginary part of the refractive index should be positive.

**s = incident beam polarization, p, s**

These buttons specify the polarization of the incident plane wave, p (parallel, also called TM) polarization, and s (perpendicular, also called TE) polarization.

**50 = mxiter, max number of iterations**

This line specifies the maximum number of iterations the user wishes to allow during the iterative process to find a solution for the feature internal field. Care should be taken whether the convergence has been met before the maximum number of iterations or not.

**1.0e-2 = Convergence Tolerance**

This is the tolerance required during the iterative process to specify when the process has reach satisfactory convergence. It is recommended that 1.0e-2 or less to be used as the error tolerance. The convergence is based on how well the dipole matrix equation is solved. This is measured by using the basic norm relation:

$$\text{Tolerance} = \|(\mathbf{B}+\mathbf{A}+\mathbf{R})\mathbf{P}-\mathbf{E}\|/\|\mathbf{E}\|$$

**0 = 0:plane wave 1:gaussian wave**

This option indicates whether the user requires the use of a Gaussian beam profile or a plane wave. This will be dependent on the ratio of the actual beam width to the characteristic length of the scattering features, a plane wave can be used when the characteristic length of the scattering system is 1/10 of the incident beam width or less.

**50 = Gaussian beam width [microns]**

This is the width of the Gaussian beam which is assumed to be focused at the surface. This is not necessary when the plane wave is designated.

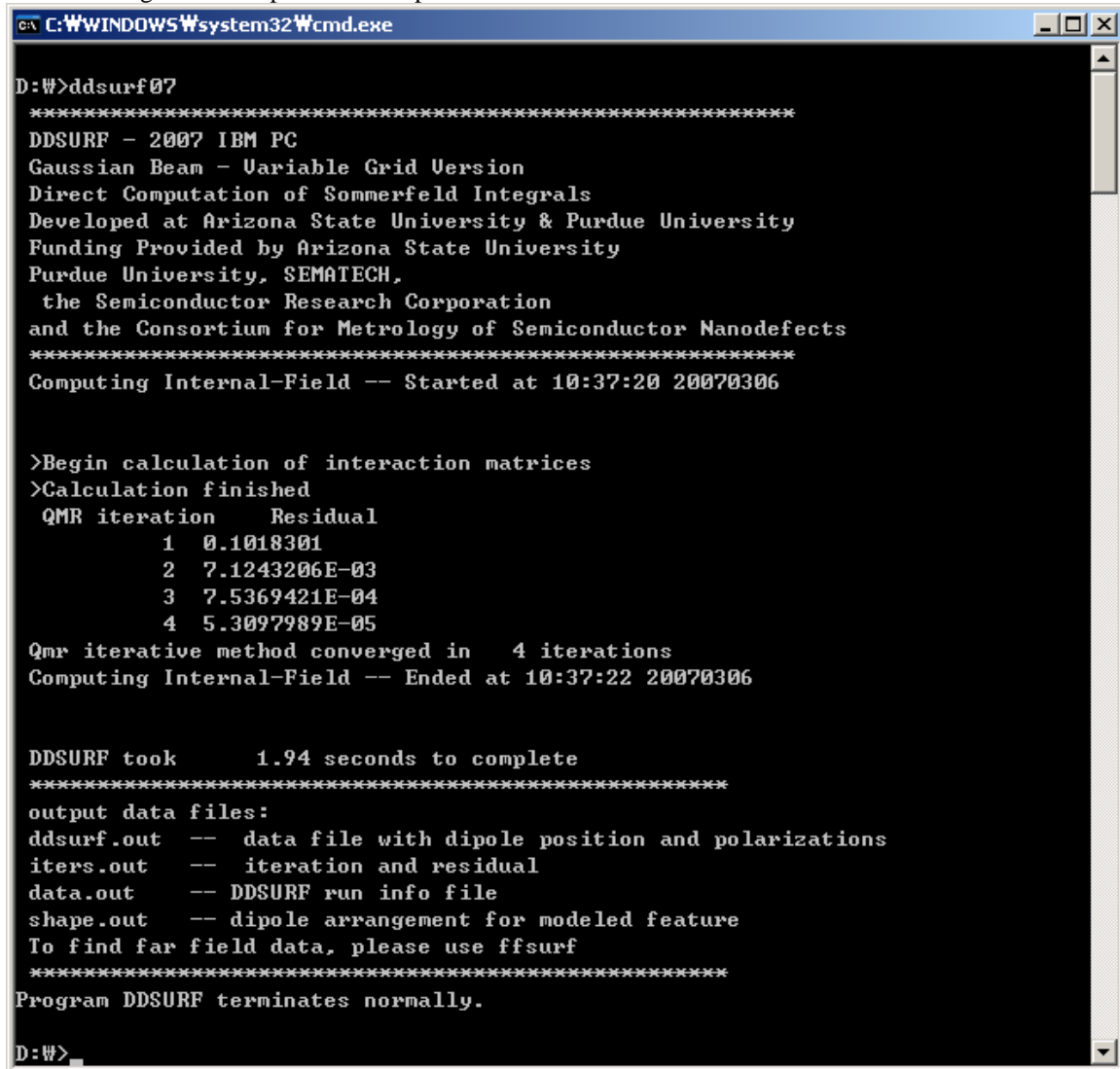
**0 0 = x,y position of center of Gauss beam [microns]**

This line specifies the center position, in microns, of the Gaussian beam based on the 0,0 position designated when developing the lattice array for the scattering feature. Special care should be taken with the positioning, considering the location of the scattering feature. When 0, 0 and desired that the beam hit the center of the feature, ensure that the center of the feature is set at x = 0 and y = 0, this is attained by viewing the input *shape.dat* file. If the center of the shape is not at 0, 0, the beam should be positioned appropriately.

## IV.2.2 Execution of DDSURF

To run DDSURF, double click *ddsurf07.exe*. When this occurs, *ddsurf07.exe* will run with parameter file *ddsurf.par*. The output screen will show some information about the execution, including the number of dipoles used, the size of the features modeled, etc. The iteration steps and the error at that step are then shown. When convergence is met, or when the maximum number of iterations has been reached, several output files will be written out to be used by FFSURF to determine the external field at the region of detection. Examples of these files are given in Appendix C

The following is an example screen output of DDSURF:



```
C:\WINDOWS\system32\cmd.exe
D:\>ddsurf07
*****
DDSURF - 2007 IBM PC
Gaussian Beam - Variable Grid Version
Direct Computation of Sommerfeld Integrals
Developed at Arizona State University & Purdue University
Funding Provided by Arizona State University
Purdue University, SEMATECH,
the Semiconductor Research Corporation
and the Consortium for Metrology of Semiconductor Nanodefects
*****
Computing Internal-Field -- Started at 10:37:20 20070306

>Begin calculation of interaction matrices
>Calculation finished
QMR iteration   Residual
      1   0.1018301
      2   7.1243206E-03
      3   7.5369421E-04
      4   5.3097989E-05
Qmr iterative method converged in 4 iterations
Computing Internal-Field -- Ended at 10:37:22 20070306

DDSURF took      1.94 seconds to complete
*****
output data files:
ddsurf.out  -- data file with dipole position and polarizations
iters.out   -- iteration and residual
data.out    -- DDSURF run info file
shape.out   -- dipole arrangement for modeled feature
To find far field data, please use ffsurf
*****
Program DDSURF terminates normally.
D:\>
```

## IV.2.3 Output files of DDSURF

The output files from DDSURF include the following:

- ddsurf.out*
- shape.out*
- data.out*
- iters.out*



*ddsurf.out* is the output file that will be used by FFSURF to compute the far-field scattering characteristics. Information in *ddsurf.out* includes the incident beam properties, number of dipoles used, dipole positions, and the internal field at these dipole positions. *shape.out* will provide the dipole configuration of the scattering features. *Data.out* describes the run that was performed, including the feature, surface, and incident field quantities. Examples of these output files are shown in Appendix C.

### IV.3 FFSURF

Once DDSURF has been executed and the internal field has been described in *ddsurf.out*, the next step is to determine the external scattered field of interest. We do this by using the FFSURF with *ffsurf.par* parameter file. The following is a screen shot of the *ffsurf.par* file.

```

2 = surface type 0: No surface 1: perf.cond. surface 2: real surface
2 = IPLOT, 1:x-pl,y-pl,z-pl, 2:radial, 3:user defined
1 1 1 1 1 = ION(1-5): E_x,E_y,E_z,Irrad, dCsc/dW , 0:no plot, 1:plot
1 = cross section (1=determine, 0= do not determine)
45 45 1 = x1,x2,delta x ==> x (theta) coordinate position ( um or deg)
0 1 1 = y1,y2,delta y ==> y (theta) coordinate position (um or deg)
1000 1000 1 = z1,z2,delta z ==> z (radius) coordinate position (um)

```

In the following sections, we will describe the input parameters need to describe the scattering system for use by FFSURF.

#### IV.3.1 FFSURF – Input Parameters

##### **2 = Surface type 0: No surface 1: perf. cond. surface 2: real surface**

This line specifies the surface characteristics of the problem. **0** means there is no surface, **1** means that the surface is a perfect conductor, and thus the method of images can be used to determine the scattering from the surface, **2** means that the surface is real, and the Fresnel coefficients are determined to compute the scattering from the surface.

##### **2 = IPLOT, 1:x-pl,y-pl,z-pl, 2:radial, 3:user defined**

This line specifies the geometry of the detection region of interest. If **1** is specified, then a plane either in the x, y, or z plane will be the detection region. The plane will be specified in the lines of *ffsurf.par* described later. If **2** is specified, a detection region of constant radius away from the feature will be created. A radius, and theta and phi angular positions will be specified later in *ffsurf.par*. If **3** is specified, an input file named *arb.dat* will be used to determine the geometric configuration of the detector region. The geometry should be specified in terms of microns, and the x, y, and z positions should be specified. An example of *arb.dat* is shown in the appendix. The position information in the rest of the input file will be ignored.

##### **1 1 1 1 1 = ION(1-5): E\_x,E\_y,E\_z,Irrad,dCsc/dW , 0:no plot, 1:plot**

These options indicate which data files will be created by FFSURF. If the user put **0**, the corresponding data file will not be created, but if the user put **1**, the corresponding data file will be created. The files are the electric field data files in the x, y, and z directions which are designated as *ex.dat*, *ey.dat*, *ez.dat*, the intensity distribution, Irrad, which is printed out to *irrad.dat* and the differential scattering cross-section, dCsc/dW, is printed out to *dfscat.dat*. The form in which it is printed out is the x, y, and z coordinates in

microns, and the fields are normalized to the incident field. These are the data files which can be plotted to show the scattering characteristics, or can be taken to calculate any quantity of interest from the detector region within the far-field, for example the differential scattering cross-section distribution on the ring-wedge detector region. When radial detection geometry is used, data files are also created using the theta, phi, and radius coordinate system also. These files will be called the same as before with an *r* extension. For example, the *ex.dat* file will also be printed out in theta, phi, and radius coordinates in *ex\_r.dat*. Examples of these files are given in Appendix C.

**1=scattering cross-section (1=determine, 0= do not determine)**

This line specifies if the user wants to determine the total scattering cross-section from the scattering feature and surface. When **1** is specified, the detector geometry must be radial, and to get a true total scattering cross-section, a hemisphere should be specified, see next 3 input lines. To determine the scattering cross-section, integration of the electric field on the detection is required. Thus, increasing the number of discrete detection points allows for a more accurate prediction of the cross-section. One method to determine if the prediction of the cross-section is converged is to take 2 trial cases with different number of points in the same region. The developers have found that positioning of points every 2 or 3 degrees in the theta direction and 5 to degrees in the phi direction is sufficient for a spherical particle.

**0 90 2 = x1,x2,delta x ==> x (theta) coordinate direction (um or deg)**  
**0 180 180 = y1,y2,delta y ==> y (phi) coordinate direction (um or deg)**  
**5000 5000 0 = z1,z2,delta z ==> z (radius) coordinate direction (um or deg)**

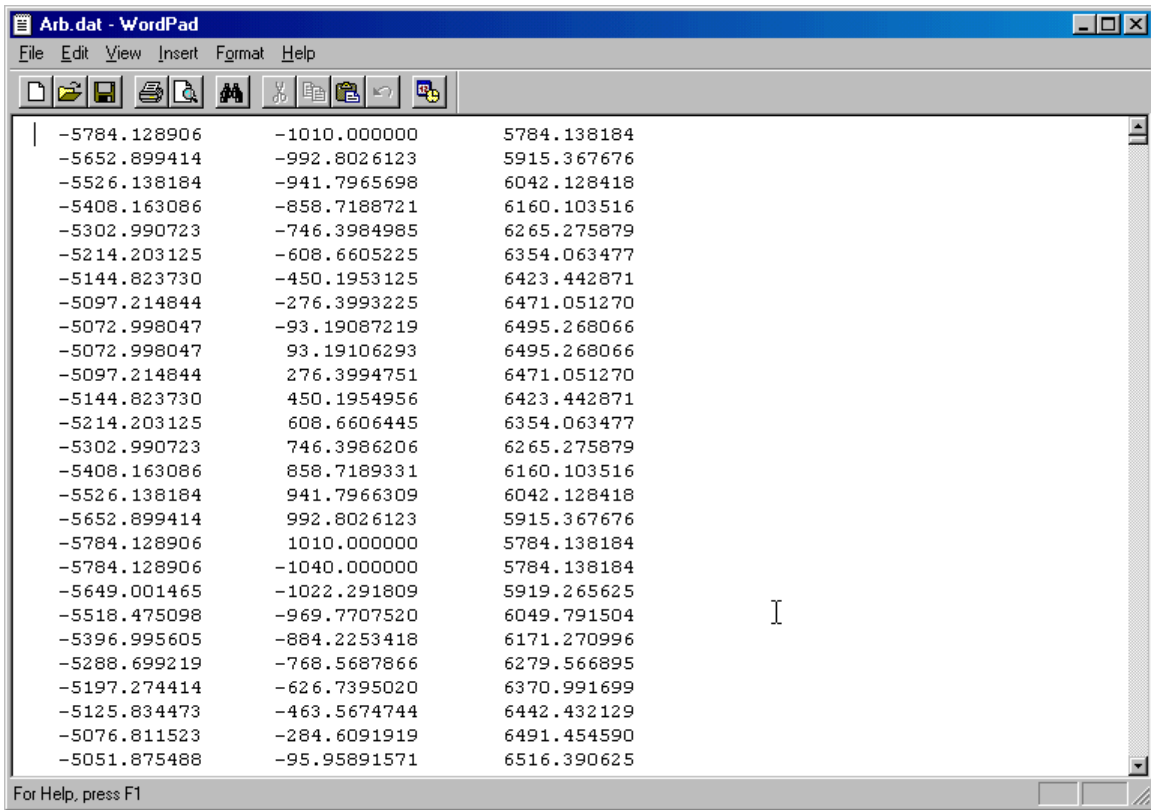
This option specifies the geometry of the detection region of interest. If the **2** is chosen for IPLOT, a detection region of constant radius from the feature will be created. A radius, and theta and phi angular positions are specified. Here, we see that the theta component, the theta angle between the point of detection and surface normal, starts at 0 (surface normal), and ends at 90 (parallel to the surface). The points are taken in steps of 2 degrees, therefore, there are 46 different theta angles considered. The phi component, the phi angle between the positive x axis and the point (counter clockwise), goes from 0 (along the positive x axis) to 180 (along the negative x axis), and takes a step on 180 degrees, thus only two phi positions are taken, 0 degs, and 180 degs. Finally, the radius is 5000 microns from the 0, 0 point of the system and the radius is constant since Radius 1 and Radius 2 are equal. Delta radius would not matter if entered for this case.

Next, we consider if the **1** is chosen for IPLOT:

**-250 -250 50 = x1,x2,delta x ==> x (theta) coordinate direction (um or deg)**  
**-250 250 50 = y1,y2,delta y ==> y (phi) coordinate direction (um or deg)**  
**0 500 50 = z1,z2,delta z ==> z (radius) coordinate direction (um or deg)**

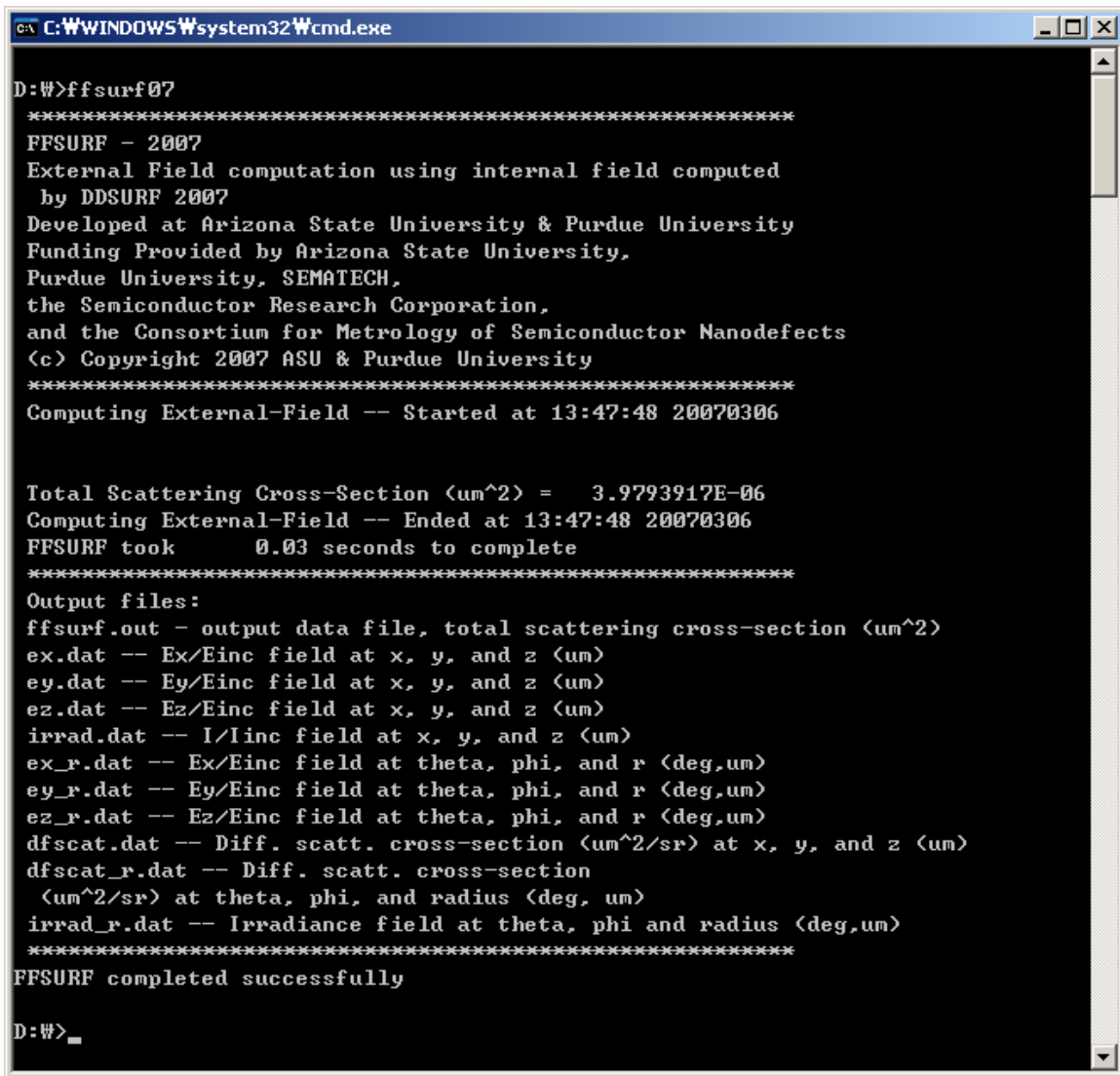
Here, we see that the specifications for the detection region changes to x, y, z coordinates. Here, a plane region has been chosen along the x = -250 μm position, and points are taken in the y direction between -250 μm and 250 μm in steps of 50 μm (11 points), and between 0 (surface) and 500 μm in the z direction (11 points), there fore this detection region has 121 points.

If **3** is chosen for IPLOT, the user will specify his own detection region by creating an input data file for FFSURF called *arb.dat*. and the user needs to use a file called *arb.dat* to indicate a user-defined geometric configuration for the region of detection. The input file *arb.dat* contains three columns, each column describes the point of detection coordinates in the x, y, and z positions respectively. The coordinates are given in microns. The following shows an example of an *arb.dat* file that was used for the ring/wedge detector used at Arizona State University:



### IV.3.2 Execution of FFSURF

After the quantities have been chosen on the FFSURF, execute FFSURF, by running `ffsurf07.exe` file.



```
C:\WINDOWS\system32\cmd.exe
D:\W>ffsurf07
*****
FFSURF - 2007
External Field computation using internal field computed
by DDSURF 2007
Developed at Arizona State University & Purdue University
Funding Provided by Arizona State University,
Purdue University, SEMATECH,
the Semiconductor Research Corporation,
and the Consortium for Metrology of Semiconductor Nanodefects
(c) Copyright 2007 ASU & Purdue University
*****
Computing External-Field -- Started at 13:47:48 20070306

Total Scattering Cross-Section (um^2) = 3.9793917E-06
Computing External-Field -- Ended at 13:47:48 20070306
FFSURF took 0.03 seconds to complete
*****
Output files:
ffsurf.out - output data file, total scattering cross-section (um^2)
ex.dat -- Ex/Einc field at x, y, and z (um)
ey.dat -- Ey/Einc field at x, y, and z (um)
ez.dat -- Ez/Einc field at x, y, and z (um)
irrada.dat -- I/linc field at x, y, and z (um)
ex_r.dat -- Ex/Einc field at theta, phi, and r (deg,um)
ey_r.dat -- Ey/Einc field at theta, phi, and r (deg,um)
ez_r.dat -- Ez/Einc field at theta, phi, and r (deg,um)
dfscat.dat -- Diff. scatt. cross-section (um^2/sr) at x, y, and z (um)
dfscat_r.dat -- Diff. scatt. cross-section
(um^2/sr) at theta, phi, and radius (deg, um)
irrada_r.dat -- Irradiance field at theta, phi and radius (deg,um)
*****
FFSURF completed successfully
D:\W>_
```

The output shows how much time was taken to execute the `ffsurf07.exe` code. The output also shows the output files that were created during the execution. The following describes the possible FFSURF files that can be created.

### IV.3.3 Output files of FFSURF

<code>ffsurf.out</code>	Contains information of the executed
<code>ex.dat</code>	Electric field (x component): position given as x, y, and z (microns)
<code>ey.dat</code>	Electric field (y component): position given as x, y, and z (microns)
<code>ez.dat</code>	Electric field (z component): position given as x, y, and z (microns)
<code>irrada.dat</code>	Irradiance distribution: position given as x, y, and z (microns)
<code>dfscat.dat</code>	Differential Scattering Cross-Section: position given as x, y, and z (microns)

Also, if a radial detection region is specified:

<i>ex_r.dat</i>	Electric field (x component): position given as <i>theta</i> , <i>phi</i> , and <i>radius</i> (degs, microns)
<i>ey_r.dat</i>	Electric field (y component): position given as <i>theta</i> , <i>phi</i> , and <i>radius</i> (degs, microns)
<i>ez_r.dat</i>	Electric field (z component): position given as <i>theta</i> , <i>phi</i> , and <i>radius</i> (degs, microns)
<i>irrad_r.dat</i>	Irradiance distribution: position given as <i>theta</i> , <i>phi</i> , and <i>radius</i> (deg, microns)

Examples of these files and their descriptions are shown in Appendix C. Discussion on how to use these files to study the scattering characteristics is also given in Appendix C.

## V REFERENCES

- Baños, A., *Dipole Radiation in the Presence of a Conducting Half-Space*, (Pergamon Press, Oxford, 1966)
- Bohren, C. F., and Huffman, D. R., *Absorption and Scattering of Light by Small Particles*, (John Wiley and Sons, 1983).
- Brenner, N. M., "Fast Fourier Transform of Externally Stored Data," *IEEE Trans. Audio Electroacoust.*, **AU-17**, 128-132 (1969).
- Burke, G. J. and Miller, E. K., "Modeling Antennas Near to and Penetrating a Lossy Interface," *IEEE Trans. Ant. and Prop.*, **AP-32(10)**, 1040-1049 (1984).
- Burke, G. J., and Poggio, A. J., *Numerical Electromagnetics Cod (NEC) - Method of Moments, Part I: Program Description - Theory*, Report UCID-18834, Lawrence Livermore Laboratory, Livermore, California (1981).
- Chen, H. C., *Theory of Electromagnetic Waves - A Coordinate-Free Approach*, (McGraw Hill, New York, 1983).
- Draine, B. T., "The Discrete-Dipole Approximation and its Application to Interstellar Graphite Grains," *Astrophys. J.*, **333**, 848-872 (1988).
- Draine, B. T., and Goodman, J., "Beyond Clausius-Mossotti: Wave Propagation on a Polarizable Point Lattice and the Discrete Dipole Approximation," *Astrophys. J.*, **405**, 685-697 (1993).
- Draine, B. T., and Flatau, P. J., "The Discrete-Dipole Approximation for Scattering Calculations," *J. Opt. Soc. Am. A*, **11**, 1491-1499 (1994).
- Freund, "Conjugate Gradient-Type Methods for Linear Systems with Complex Symmetric Coefficient Matrices," *SIAM J. Sci. Stat. Comput.*, vol 13, No1, pp. 425-448 (1992).
- Goodman, J. J., Draine, B. T., and Flatau, P. J., "Application of Fast-Fourier-Transform Techniques to the Discrete-Dipole Approximation," *Optics Letters*, **16**, No. 15, 1198-1200 (1991).
- Kozaki, S., and Sakurai, H., "Characteristics of a Gaussian Beam at a Dielectric Interface," *Journal of the Optical Society of America A*, **68**, No. 4, 508-514 (1978).
- Lager, D. L., and Lytle, R. J., *Fortran Subroutines for the Numerical Evaluation of Sommerfeld Integrals unter Anterem*, Rep. UCRL-51821 (Lawrence Livermore Laboratory, Livermore, Calif., 1975).
- Mohsen, A., "On the Evaluation of Sommerfeld Integrals," *IEE PROC.*, **129(4)**, 177-182 (1982).
- Nebeker, B. M., Starr, G. W., and Hirleman, E. D., "Light Scattering from Patterned Surfaces and Particles on Surfaces," *Optical Characterization Techniques for High Performance Microelectronic Device Manufacturing II*, J. K. Lowell, R. T. Chen, J. P. Mathur Eds., Proc. SPIE 2638, 274-284, (1995).
- Purcell, E. M. and Pennypacker, C. R., "Scattering and Absorption of Light by Nonspherical Dielectric Grains," *Astrophys. J.*, **186**, 705-714 (1973).
- Rahola, J., "Solution of Dense Systems of Linear Equations in the Discrete-Dipole Approximation," *SIAM Journal on Scientific*, **17**, to be published (1996).

Schmehl, R., "The Coupled-Dipole Method for Light Scattering for Light Particles on Plane Surfaces," (Department of Mechanical and Aerospace Engineering, Arizona State University and the Institut für Thermische Strömungsmaschinen, Universität Karlsruhe, Germany, 1994).

Schmehl, R., Nebeker, B. M., and Hirleman, E. D., "The Coupled-Dipole Method for Scattering by Particles on Surfaces Using a Two-Dimensional Fast Fourier Transform Technique," *J. Opt. Soc. Am. A*, to be reviewed (1997).

Taubenblatt, M. A., and Tran, T. K., "Calculation of Light Scattering from Particles and Structure by the Coupled-Dipole Method," *J. Opt. Soc. Am. A*, **10**, 912-919 (1993).

## APPENDIX A

In this appendix, we show the code setup, including the block diagrams used by the executable files of DDSURF.

### A.1 DDSURF – File List

*CREATETAR07.EXE* – The Fortran 90 based code which computes the dipole lattice array used to describe the scattering feature. This executable was developed using Digital/Microsoft Visual Fortran 5.0.

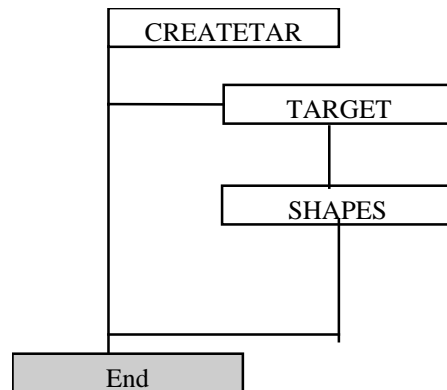
*DDSURF07.EXE* – The Fortran 90 based code which computes the internal field of the scattering feature. This executable was developed using Digital/Microsoft Visual Fortran 5.0.

*FFSURF07.EXE* – The Fortran 90 based code which computes the external far-field scattering based on the internal field of the scattering feature found by DDSURF07. This executable was developed using Digital/Microsoft Visual Fortran 5.0.

### A.2 CREATETAR - Code Setup and Subroutine List

#### A.2.1 Block Diagram of CREATETAR

The following shows a block diagram describing the subroutine structure of CREATETAR and the subroutine listing:

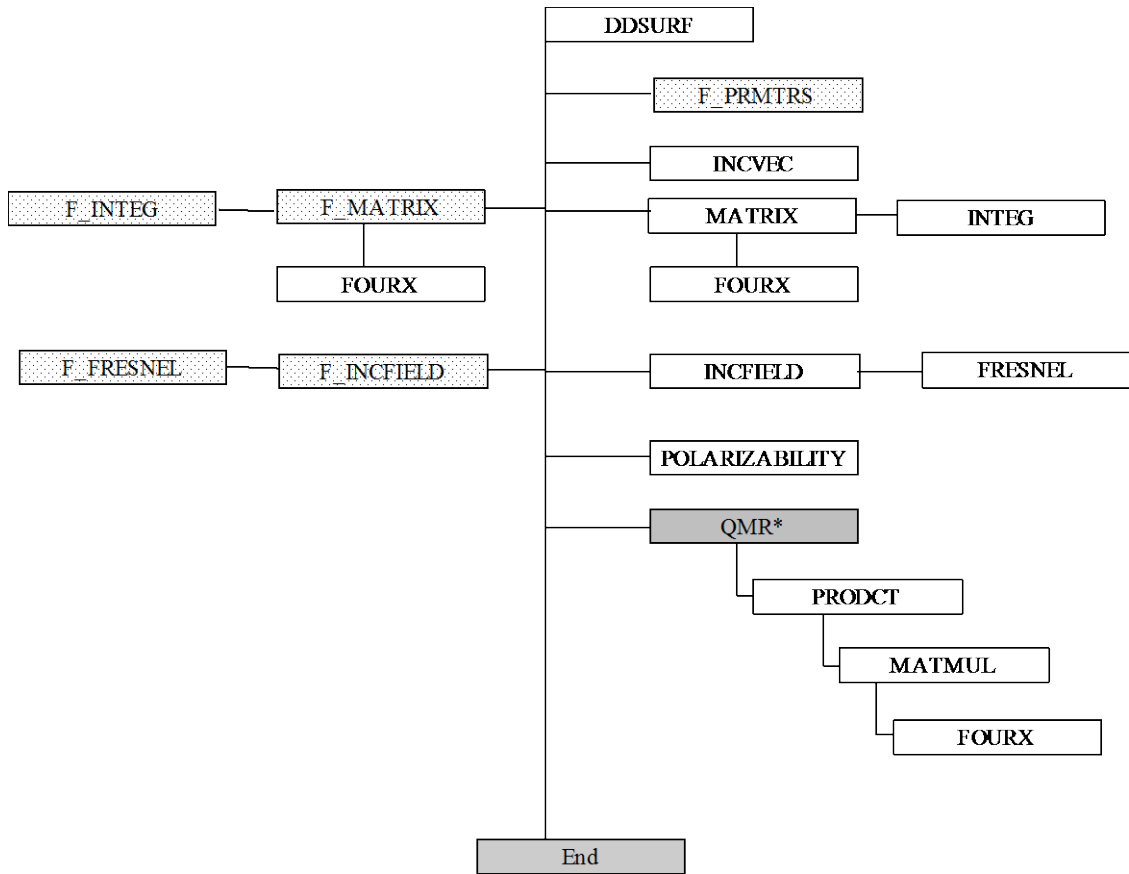


### A.3 DDSURF - Code Block Diagram

#### A.3.1 Block Diagram of DDSURF

The following is the block diagram for DDSURF. The dotted subroutines are new routines which take the film surface into account. For example, F\_MATRIX considers the film surface while MATRIX does not. The decision to which subroutine is used are made within the code.

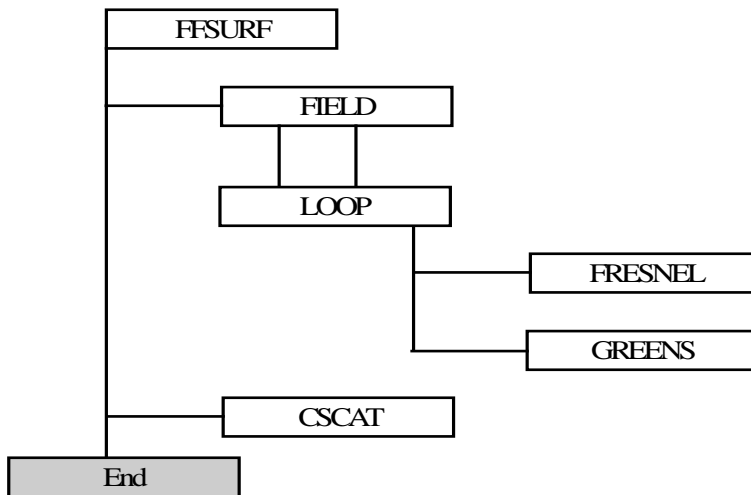




**A.4 FFSURF - Code Block Diagram**

**A.4.1 Block Diagram of FFSURF**

The following is the block diagram for the subroutines of FFSURF.



## Appendix B

In this appendix, information about various aspects about the discrete dipole approximation is discussed. The purpose of this appendix is to give the user a short background which will lead to further study in other references about the discrete dipole approximation and its components.

### B.1 Dyadic Green's Function

The dyadic Green's function which is used to compute the electric field radiated from a dipole can be represented in the form (Chen, 1983)

$$\overline{G}(r, r') = \left[ \overline{I} + \frac{\nabla\nabla}{k^2} \right] g(R), \quad (\text{B1})$$

$$\overline{G}(r, r') = \overline{G}(R) = g(R) \left\{ \left[ 1 - (kR)^{-2} + i(kR)^{-1} \right] \overline{I} - \left[ 1 - 3(kR)^{-2} + i3(kR)^{-1} \right] \hat{R}\hat{R} \right\}, \quad (\text{B2})$$

where we introduce

$$R = r - r', \quad (\text{B3a})$$

$$R = |r - r'|, \quad (\text{B3b})$$

$$\hat{R} = R / R, \quad (\text{B3c})$$

$$g(R) = (4\pi R)^{-1} \exp(ikR), \quad (\text{B3d})$$

for discrete representation we designate the dyadic Green's function as

$$\overline{G}(r, r') = \overline{G}_{ij} \quad (\text{B4})$$

where  $r = r_i$  and  $r' = r_j$  and  $\nabla\nabla$  is the dyadic operator.

### B.2 Reflection Coefficients

The following are definitions of the Fresnel scattering coefficients used in Eq. 15 to determine scattering from a surface.

$$R^{TE} = \frac{k_{2z}\mu_1 - k_{1z}\mu_2}{k_{2z}\mu_1 + k_{1z}\mu_2} \quad (\text{B5})$$

$$R^{TM} = \frac{k_{2z}\varepsilon_1 - k_{1z}\varepsilon_2}{k_{2z}\varepsilon_1 + k_{1z}\varepsilon_2} \quad (\text{B6})$$

where  $k_z$  is the wave number in the  $z$  direction,  $\mu$  is the magnetic permeability, and  $\varepsilon$  is the dielectric permittivity, the subscripts 1 indicates the surface region, and 2 indicates the region above the surface .

### B.3 Sommerfeld Integral Terms

Recall that the dipole radiation reflected from a dielectric surface has the following form:

$$\bar{E}_{reflected,i} = \sum_{j=1}^N \left( \bar{S}_{ij} + \frac{k_2^2}{\epsilon_0} \frac{k_1^2 - k_2^2}{k_1^2 + k_2^2} \bar{G}_{ij}^I \right) \cdot \bar{P}_j \quad (B7)$$

The Sommerfeld integral expressions are contained in the matrix  $\bar{S}_{ij}$ . The components of this matrix are group of integrals that consider the vertical and horizontal dipole components are determined numerically in the *table.f* subroutine group

#### **B.4 Gaussian Beam Equations**

The following is the development of the Gaussian beam equations by Kozaki and Sakurai for the traverse component of the electromagnetic field (x, if the beam travels in the y-z plane).

For example, the x-component of the Gaussian field in integral form is:

##### Incident Field

$$E_x^i(y, z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} E_i(\alpha) \exp[-jyp - j(z + z_0)(k_1^2 - p^2)^{1/2}] d\alpha \quad (B8)$$

##### Reflected Field

$$E_x^r(y, z) = \frac{1}{2\pi} \int_{-\infty}^{\infty} E_i(\alpha) R(\alpha) \exp[-jyp - j(z + z_0)(k_1^2 - p^2)^{1/2}] d\alpha \quad (B9)$$

where

$$p = \alpha \cos \theta + \sin \theta (k_1^2 - \alpha^2)^{1/2}$$

Where the incident beam was in the z-y plane.

A Taylor series plane wave expansion about  $\alpha = 0$  was used to describe the integrals. The form for the x component of the electric field as given by Kozaki and Sakurai were found to be:

##### Incident Field

$$E_x^i(y_i, z_i) \cong \frac{1}{(1 - j2a^2 z_i / k_1)^{1/2}} \exp\left(-\frac{(ay_i)^2}{(1 - j2a^2 z_i / k_1)} - jk_1 z_i\right) \quad (B10)$$

##### Reflected Field

$$E_x^r(y_r, z_r) \cong \left\{ \begin{array}{l} \left[ \frac{R(0)}{(1 + j2a^2 z_r / k_1)^{1/2}} - j \frac{2a^2 R'(0) y_r}{(1 + j2a^2 z_r / k_1)^{3/2}} + \right. \\ \left. \frac{a^2 R''(0)}{(1 + j2a^2 z_r / k_1)^{3/2}} \left( 1 - \frac{2(ay_r)^2}{1 + j2a^2 z_r / k_1} \right) \right] \\ \times \exp\left(-\frac{(ay_r)^2}{(1 + j2a^2 z_r / k_1)} + jk_1 z_r\right) \end{array} \right\} \quad (B11)$$

Where the reflection coefficient, for example for the s-polarized waves can be denoted as:

$$R(0) = \frac{n \cos \theta - (1 - n^2 \sin^2 \theta)^{1/2}}{n \cos \theta + (1 - n^2 \sin^2 \theta)^{1/2}}$$

$$R'(0) = \frac{2 \sin \theta}{k_2 (1 - n^2 \sin^2 \theta)^{1/2}} R(0) \quad (\text{B12})$$

$$R''(0) = \frac{2 \left[ \cos \theta + 2n \sin^2 \theta (1 - n^2 \sin^2 \theta)^{1/2} \right]}{k_2^2 n \left[ (1 - n^2 \sin^2 \theta)^{1/2} \right]^3} R(0)$$

where

$$n = \frac{n_i}{n_t}, \text{ and } u = \frac{\cos \theta}{\cos \theta_t}, \quad v = \frac{n \cos^2 \theta}{\cos^3 \theta_t} z + \frac{z_0}{\cos \theta} + \frac{\sin \theta}{\cos \theta_t} y_t$$

### **B.5 Polarizability Modeling - Lattice Dispersion Relation and Elliptical Dipole Modeling**

In DDSURF 1/2007, two different methods may be used to determine the dipole polarizability,  $\alpha_i$ . If the lattice spacing is uniform, DDSURF uses the method called the Lattice Dispersion Relation (Draine *et al.*, 1993). Draine used the basis of modeling a polarizability so that a lattice with finite spacing could mimic a continuum. The expression was determined to be

$$\alpha \approx \frac{\alpha^{(0)}}{1 + (\alpha_i^{(0)} / d^3) \left[ (b_1 + m^2 b_2 + m^3 b_3 S)(k_0 d)^2 - (2/3)i(k_0 d)^3 \right]} \quad (\text{B13})$$

where

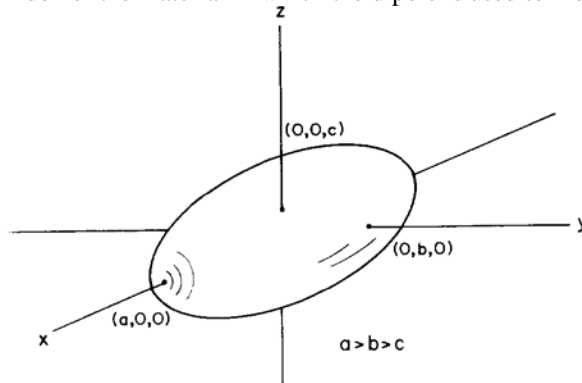
$$\alpha^{(0)} \equiv \frac{3d^3}{4\pi} \left( \frac{m_i^2 - 1}{m_i^2 + 2} \right) \text{ for a spherical dipole} \quad (\text{B14a})$$

and

$$b_1 = -(4\pi/3)^{1/2}, \quad b_2 = 0.1648469, \quad b_3 = -1.7700004 \quad (\text{B14b})$$

$$S \equiv \sum_{j=1}^3 (a_j e_j)^2, \quad (\text{B14c})$$

where  $a$  and  $e$  are unit vectors defining the incident direction and the polarization state, respectively, and  $m$  is the relative refractive index of the material in which the dipole is used to model.



Example of an ellipsoidal dipole.

If the dipole spacing used in DDSURF is non-uniform, a different method of determining the dipole polarizability must be used as the Lattice Dispersion Relation has not been developed for elliptical dipoles. When the lattice spacing is non-uniform, the shape of the dipole becomes ellipsoidal as shown in Figure 9.7.1, a more general approach of Bohren and Huffman (1983) based on the Clausius-Mossotti relation.

The dipole polarizability now becomes dependent on the principal axes. In terms of dielectric permittivity, Bohren and Huffman (1983) give the dipole polarizability of an ellipsoidal dipole as:

$$\alpha_j = abc \frac{(\varepsilon - \varepsilon_m)}{3\varepsilon_m + 3L_j(\varepsilon - \varepsilon_m)}, j = 1,2,3, \quad (\text{B15})$$

where  $a > b > c$  are the semi-axes and  $L_j$  are geometrical factors which add up to equal one, and are given by for the special cases of a prolate spheroid and an oblate spheroid:

For a prolate spheroid (b=c):

$$L_1 = \frac{1-e^2}{e^2} \left( -1 + \frac{1}{2e} \ln \frac{1+e}{1-e} \right) \quad e^2 = 1 - \frac{b^2}{a^2} \quad (\text{B16a})$$

$$L_2 = L_3 \quad (\text{B16b})$$

For an oblate spheroid (a=b):

$$L_1 = \frac{g(e)}{2e^2} \left[ \frac{\pi}{2} - \tan^{-1} g(e) \right] - \frac{g^2(e)}{2} \quad L_2 = L_3 \quad (\text{B16c})$$

$$g(e) = \left( \frac{1-e^2}{e^2} \right)^{1/2}, \quad e^2 = 1 - \frac{c^2}{a^2} \quad (\text{B16d})$$

The next step is determine the effective electric permittivity if we were to extend the electric field of the ellipsoid dipole to the entire rectangular region that the lattice spacing represents. Singham et. al. gives method to determine  $\varepsilon$ , based on the particle permittivity, and the dipole spacing. By the Maxwell-Garnett theory for effective medium, we find  $\varepsilon$  based on the following equations:

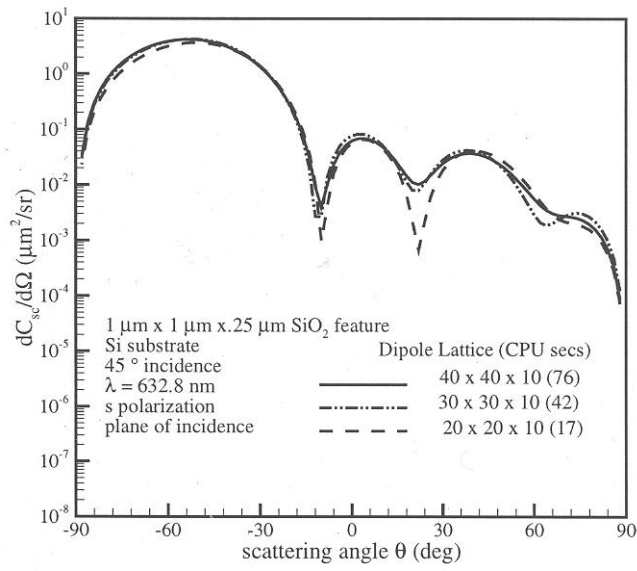
$$\varepsilon_{av} = \frac{(1-f)\varepsilon_m + f\beta\varepsilon}{1-f + f\beta} \quad (\text{B17})$$

$$\beta = \frac{\varepsilon_m}{3} \sum_{j=1}^3 \frac{1}{\varepsilon_m + L_j(\varepsilon - \varepsilon_m)} \quad (\text{B18})$$

Where  $\varepsilon_{av}$  is the electric permittivity of the scattering medium and  $f$  is the volumetric ratio of the ellipsoid to the rectangular block it is to represent.

DDSURF determines the effective permittivity by solving Eqs. (B15 through B18) iteratively, using an initial guess for  $\varepsilon$  to be slightly higher than  $\varepsilon_{av}$ , as we expect a rectangular volume to scatter more than an ellipsoid volume. Because of the mathematical nature of the equations, computation of non-spherical dipoles has been restricted to particles in which the magnitude of the refractive index is less than 2.1. If the refractive index is higher than this, convergence during the iterative procedure cannot be attained. Therefore, DDSURF restricts the computation of features with refractive indexes higher than 2.1 to use only uniformly spaced lattice points.

Consideration of the accuracy of the results should be taken when using a non-uniform grid spacing. At this point DDSURF 1/2007 is restricted to the lattice spacing in the x and y directions to be equal, as well as the ratio of the lengths in the x and y directions to the lattice length in the z direction should be restricted to approximately no greater than 1.5. Figures 9.7.2(a)-(b) show comparisons of the differential scattering cross-section at a constant radius along the plane of incidence around a rectangular feature, where the lattice spacing in the z direction was modified while the geometry of the scatterer was maintained to be the same.



Example of comparisons of scattered irradiance distribution around a constant radius in the plane of incidence.  $1 \mu\text{m} \times 1 \mu\text{m} \times 0.25 \mu\text{m}$  rectangular feature.

We see from the graphs that when the ratio of the x,y lattice spacing to the z lattice spacing is 2.0, the comparisons tend to deviate from the original, uniform dipole spacing configuration.

## APPENDIX C

In the following section, we will present an example problem, of scattering from a rectangular feature on a surface. We will show the input and output files of DDSURF and FFSURF, as well as some plots that exhibit the scattering characteristics of the problem.

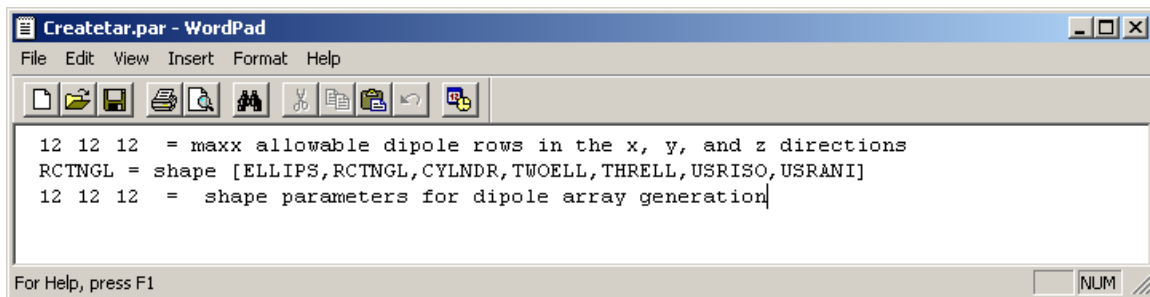
Let us consider the following case:

Scattering from a  $0.3 \mu\text{m} \times 0.3 \mu\text{m} \times 0.3 \mu\text{m}$   $\text{SiO}_2$  rectangular feature. The incident beam is a Helium - Neon laser beam with a wavelength of  $0.6328 \mu\text{m}$  and s - polarized, and the beam is assumed to act as a plane wave. The beam is 45 degrees incident from the normal of the surface.

### C.1 CREATETAR

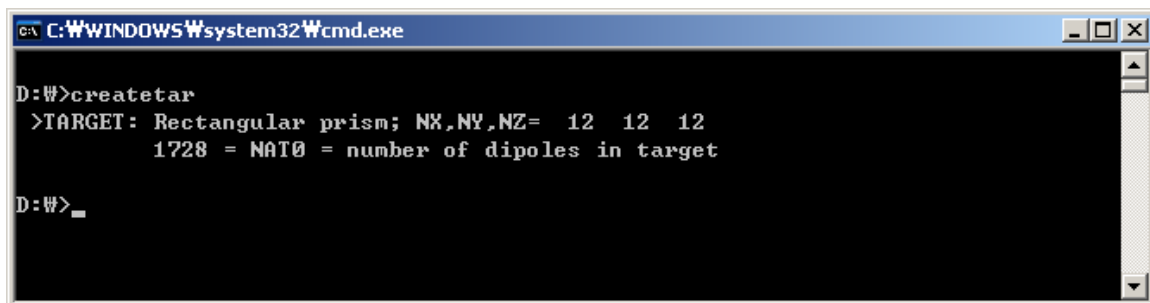
#### C.1.1 CREATETAR Execution

To create the rectangular feature and create the *shape.dat* file needed by DDSURF, the input parameters for CREATETAR should look as the following:



```
12 12 12 = maxx allowable dipole rows in the x, y, and z directions
RCTNGL = shape [ELLIPS,RCTNGL,CYLNDR,TWOELL,THRELL,USRISO,USRANI]
12 12 12 = shape parameters for dipole array generation
```

When CREATETAR is executed the *shape.dat* file to be used by DDSURF is created. The output screen should look as follows:



```
C:\WINDOWS\system32\cmd.exe
D:\>createtar
>TARGET: Rectangular prism; NX,NY,NZ= 12 12 12
1728 = NAT0 = number of dipoles in target
D:\>
```

### C.2 DDSURF

#### C.2.1 DDSURF Execution

Now that *shape.dat* has been created by CREATETAR, we will use DDSURF to determine the internal scattering field of the cubic scatter. The following shows the input parameters which should be input. Note that since our cube is  $12 \times 12 \times 12$  dipole rows, and we wish to model a  $0.3 \mu\text{m} \times 0.3 \mu\text{m} \times 0.3 \mu\text{m}$   $\text{SiO}_2$  rectangular feature, the dipole spacing should be  $.025$ . The refractive index of  $\text{SiO}_2$  at  $632.8 \text{ nm}$  is  $(1.46, 0)$  and the refractive index of Si at  $632.8 \text{ nm}$  is  $(3.88, 0.023)$ .

```

ddsurf.par - WordPad
File Edit View Insert Format Help
[Icons]
'***** INPUT FILE FOR PROGRAM DDSURF*****'
2      = 0:no surface, 1:perf.cond. surface, 2:dielectric surface
0      = H, is z value of first dipole z-layer (microns) (H=0. --> on surf.)
1      = 0 means enter radius, 1 means enter lattice spacing
0.025,0.025 = effective radius or lattice spacing, (x and y) and z directions [microns]
.6328  = wavelength of incident wave [microns]
45.0   = THETA, angle between z-axis and incident wave vector [deg]
0.0    = PHI, angle between +x-axis and incident plane [deg]
1      = number of different dielectric constants in feature
(1.46,0) = complex refractive index for feature material No.1 [IM>=0]
0      = film thickness [microns]
(1.0,0) = complex refractive index for film [IM>=0]
(3.88,0.023) = complex refractive index for material below surface [IM>=0]
s      = incident beam polarization, p,s
200    = mxiter, max number of iterations
1.0e-3 = Convergence Tolerance = |(B+A+R)P-E|/|E|
0      = 0:plane wave 1:gaussian wave
50     = Gaussian beam width [microns]
0 0    = x,y position of center of Gaussian beam [microns]
For Help, press F1
NUM

```

The code is executed, and the screen output is as follows:

```

C:\WINDOWS\system32\cmd.exe
D:\>ddsurf07
*****
DDSURF - UERSION 1/2007 IBM PC
Gaussian Beam - Variable Grid Uersion
Substrate and Film Uersion
Developed at Arizona State University & Purdue University
Funding Provided by Arizona State University, SEMATECH,
the Semiconductor Research Corporation
and the Consortium for Metrology of Semiconductor Nanodefects
*****
Computing Internal-Field -- Started at 14:31:15 20070306

>Begin calculation of interaction matrices
>Calculation finished
  QMR iteration   Residual
      1  0.2930935
      2  3.8357429E-02
      3  4.2966660E-03
      4  5.7742797E-04
Qmr iterative method converged in 4 iterations
Computing Internal-Field -- Ended at 14:31:18 20070306

DDSURF took 3.20 seconds to complete
*****
output data files:
ddsurf.out -- data file with dipole position and polarizations
iters.out  -- iteration and residual
data.out   -- DDSURF run info file
shape.out  -- dipole arrangement for modeled feature
To find far field data, please use ffsurf
*****
Program DDSURF terminates normally.
D:\>_

```



From this output, we can tell how many iterations the process took to reach convergence, the time in which the project took to converge, and which output files have been created by DDSURF.

### C.2.2 DDSURF output files

In this section, we will describe the output files which are created with the execution of DDSURF. These files will include:

- ddsurf.out – data file with dipole position and polarization (to be used by FFSURF)
- iters.out – iterations and residuals of the process
- data.out – information file for the run just completed by DDSURF
- shape.out – dipole arrangement for modeled feature

#### ddsurf.out

The most important data file created by execution of DDSURF is *ddsurf.out*. *ddsurf.out* contains the dipole polarization (dipole moment) information at each dipole which is used by FFSURF to determine the external scattered field. Below shows an example of *ddsurf.out* for the example case considered. The first data line specifies the number of dipoles used in the extended target, and the number in the original target (rectangular array). Since the feature is rectangular, the two numbers are the same. If the feature was ellipsoidal, the first number would be greater than the second. The second data line shows the x, y, and z wave numbers in 1/um units. The third data line shows the vector wave number and the dipole position with respect to the surface. The number should be minus half the z lattice spacing (reasons are within FFSURF coding). The third line is the x, y, and z lattice spacing. The next line indicates the incident beam wavelength, film thickness, film dielectric constant (set at (1.0,0) is none present), and the substrate dielectric constant). The remainder of the file shows the dipole position in the x, y, and z dipole units, and the x, y, and z dipole moments respectively. These values will be used by FFSURF to determine the external far-field scattering.

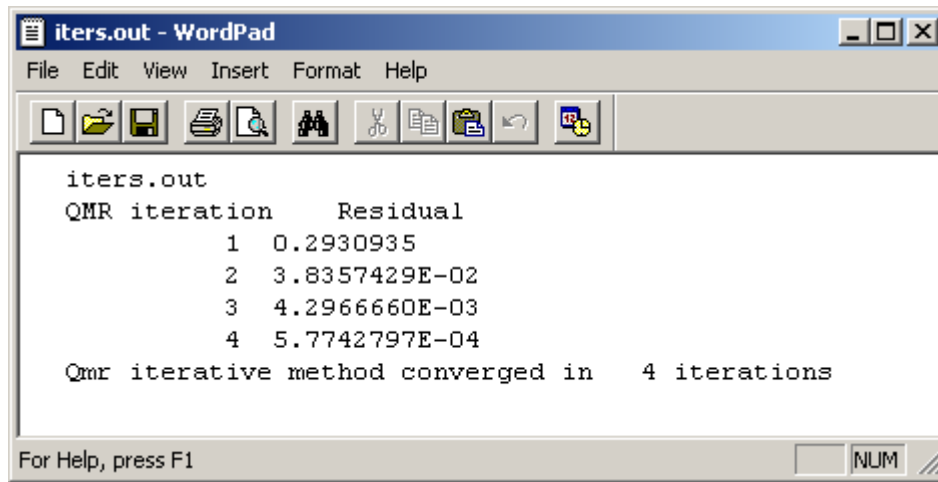
```

ddsurf.out
# of dipoles in extended target, # of dipoles in original target
1728 1728
wave numbers: x, y, z coordinates (1/um)
-7.02099132538 0.00000000000 -7.02099132538
Wave number, feature, height above top surface
9.929180 -1.2500000E-02
x y and z lattice spacing (um)
2.5000000E-02 2.5000000E-02 2.5000000E-02
wavelength (um), thickness (um), dielectric constant: film, substrate
0.6328000 0.0000000E+00 (1.000000,0.000000E+00) (15.05387,0.1784800)
dipole x,y,(z-height above surface) dipole moment (x,y,z)
1 1 1 (-0.45804E-07, 0.14538E-06) (-0.60260E-06,-0.58536E-07) (-0.93177E-06, 0.57598E-06)
2 1 1 ( 0.10549E-07, 0.14568E-06) (-0.58511E-06, 0.31047E-08) (-0.97835E-06, 0.65187E-06)
3 1 1 ( 0.52936E-07, 0.12391E-06) (-0.56153E-06, 0.70566E-07) (-0.96182E-06, 0.71198E-06)
4 1 1 ( 0.86076E-07, 0.93575E-07) (-0.51261E-06, 0.13850E-06) (-0.88728E-06, 0.74899E-06)
5 1 1 ( 0.11094E-06, 0.59018E-07) (-0.44173E-06, 0.20209E-06) (-0.76968E-06, 0.76159E-06)
6 1 1 ( 0.12684E-06, 0.23787E-07) (-0.35323E-06, 0.25681E-06) (-0.62413E-06, 0.74910E-06)
7 1 1 ( 0.13238E-06,-0.90129E-08) (-0.25194E-06, 0.29826E-06) (-0.46523E-06, 0.71186E-06)
8 1 1 ( 0.12654E-06,-0.36699E-07) (-0.14314E-06, 0.32276E-06) (-0.30648E-06, 0.65202E-06)
9 1 1 ( 0.10931E-06,-0.56999E-07) (-0.32774E-07, 0.32791E-06) (-0.16027E-06, 0.57410E-06)
10 1 1 ( 0.81811E-07,-0.68112E-07) ( 0.72613E-07, 0.31298E-06) (-0.38049E-07, 0.48897E-06)
11 1 1 ( 0.45179E-07,-0.68404E-07) ( 0.16662E-06, 0.27892E-06) ( 0.49430E-07, 0.39588E-06)
12 1 1 ( 0.27382E-08,-0.62824E-07) ( 0.25694E-06, 0.23253E-06) ( 0.88150E-07, 0.32308E-06)
1 2 1 (-0.17894E-07, 0.12066E-06) (-0.75300E-06,-0.14453E-06) (-0.77800E-06, 0.48897E-06)
2 2 1 ( 0.14346E-07, 0.12503E-06) (-0.75320E-06,-0.63764E-07) (-0.82219E-06, 0.53013E-06)
3 2 1 ( 0.44758E-07, 0.10918E-06) (-0.73179E-06, 0.19567E-07) (-0.82414E-06, 0.56913E-06)
4 2 1 ( 0.70486E-07, 0.83924E-07) (-0.67713E-06, 0.10556E-06) (-0.77767E-06, 0.59463E-06)
5 2 1 ( 0.90705E-07, 0.54083E-07) (-0.59309E-06, 0.18873E-06) (-0.69393E-06, 0.60401E-06)

```

### iters.dat

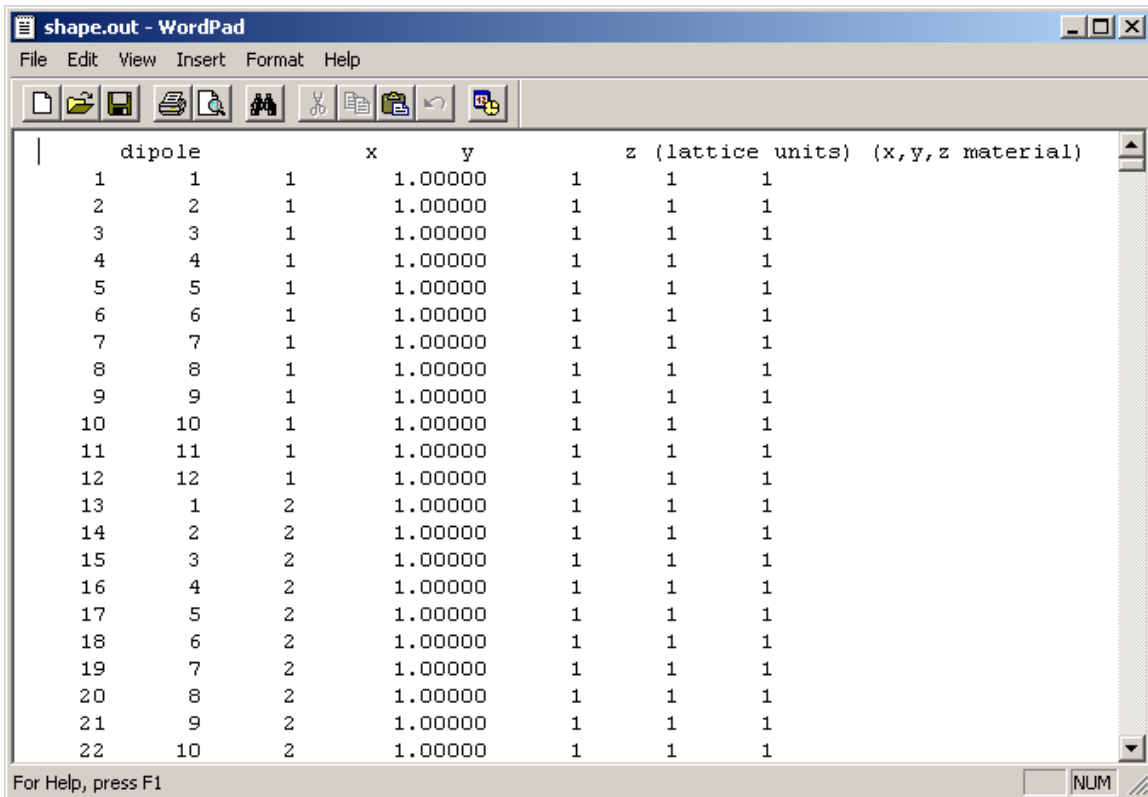
The following shows the *iters.out* data file which shows the iteration count and the error associated with that iteration step.



```
iters.out
QMR iteration      Residual
      1  0.2930935
      2  3.8357429E-02
      3  4.2966660E-03
      4  5.7742797E-04
Qmr iterative method converged in 4 iterations
```

### shape.out

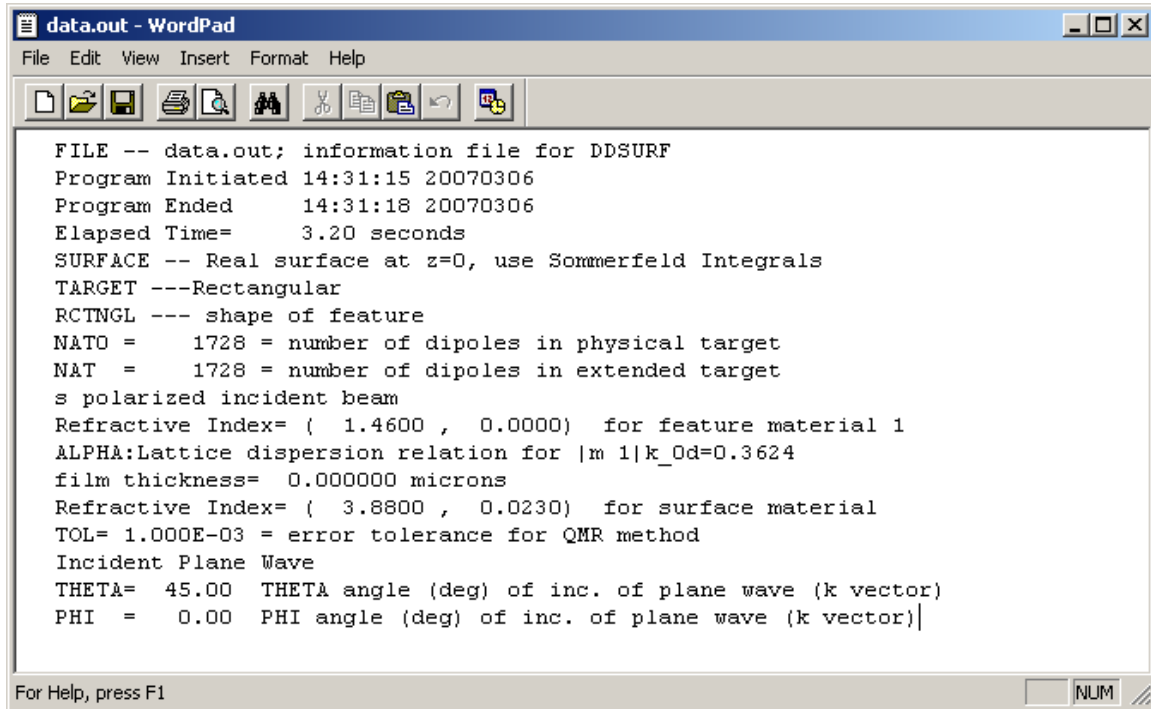
The following shows the *shape.out* file which is created with the execution of DDSURF. This contains information of the x, y, and z dipole positions as well as the material designations. This is performed to ensure that the correct shape is being read by DDSURF and FFSURF.



```
shape.out
dipole      x      y      z (lattice units) (x,y,z material)
1  1  1  1.00000  1  1  1
2  2  1  1.00000  1  1  1
3  3  1  1.00000  1  1  1
4  4  1  1.00000  1  1  1
5  5  1  1.00000  1  1  1
6  6  1  1.00000  1  1  1
7  7  1  1.00000  1  1  1
8  8  1  1.00000  1  1  1
9  9  1  1.00000  1  1  1
10 10  1  1.00000  1  1  1
11 11  1  1.00000  1  1  1
12 12  1  1.00000  1  1  1
13  1  2  1.00000  1  1  1
14  2  2  1.00000  1  1  1
15  3  2  1.00000  1  1  1
16  4  2  1.00000  1  1  1
17  5  2  1.00000  1  1  1
18  6  2  1.00000  1  1  1
19  7  2  1.00000  1  1  1
20  8  2  1.00000  1  1  1
21  9  2  1.00000  1  1  1
22 10  2  1.00000  1  1  1
```

## data.out

The following shows the *data.out* file which shows information about the run that was just completed. This is performed as a check for the user to make sure the correct set of parameters was used by DDSURF.



```
FILE -- data.out; information file for DDSURF
Program Initiated 14:31:15 20070306
Program Ended 14:31:18 20070306
Elapsed Time= 3.20 seconds
SURFACE -- Real surface at z=0, use Sommerfeld Integrals
TARGET ---Rectangular
RCTNGL --- shape of feature
NATO = 1728 = number of dipoles in physical target
NAT = 1728 = number of dipoles in extended target
s polarized incident beam
Refractive Index= ( 1.4600 , 0.0000) for feature material 1
ALPHA:Lattice dispersion relation for |m 1|k_Od=0.3624
film thickness= 0.000000 microns
Refractive Index= ( 3.8800 , 0.0230) for surface material
TOL= 1.000E-03 = error tolerance for QMR method
Incident Plane Wave
THETA= 45.00 THETA angle (deg) of inc. of plane wave (k vector)
PHI = 0.00 PHI angle (deg) of inc. of plane wave (k vector)
```

### C.3 FFSURF

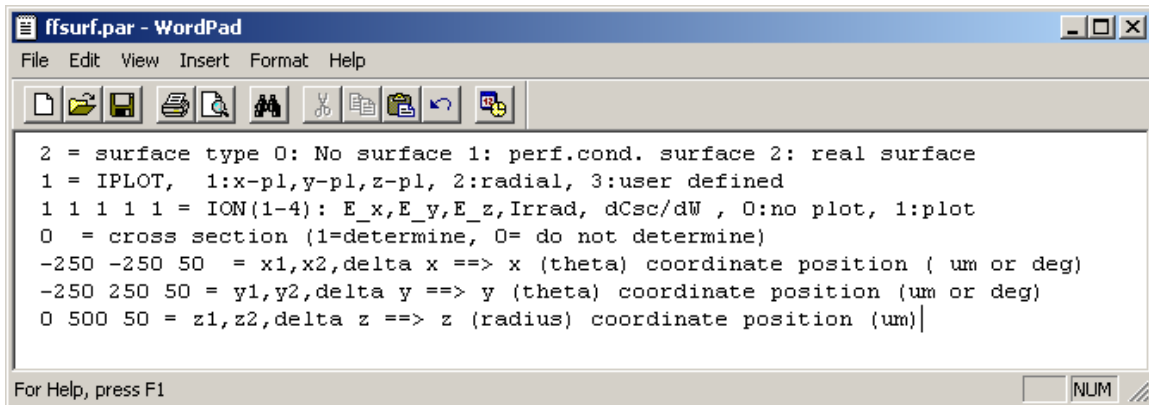
CREATETAR created the dipole arrangement for the scatterer, and DDSURF determined the internal scattering field of the rectangular scatterer for the set of input parameters. Now we show examples of determining the external field for the example case considered. We will consider three different regions: (1) a region in a plane with constant  $x$  values; (2) A hemispherical region; and (3) a region along the plane of incidence.

#### **C.3.1 Field Distribution along a plane detection region of constant $x$**

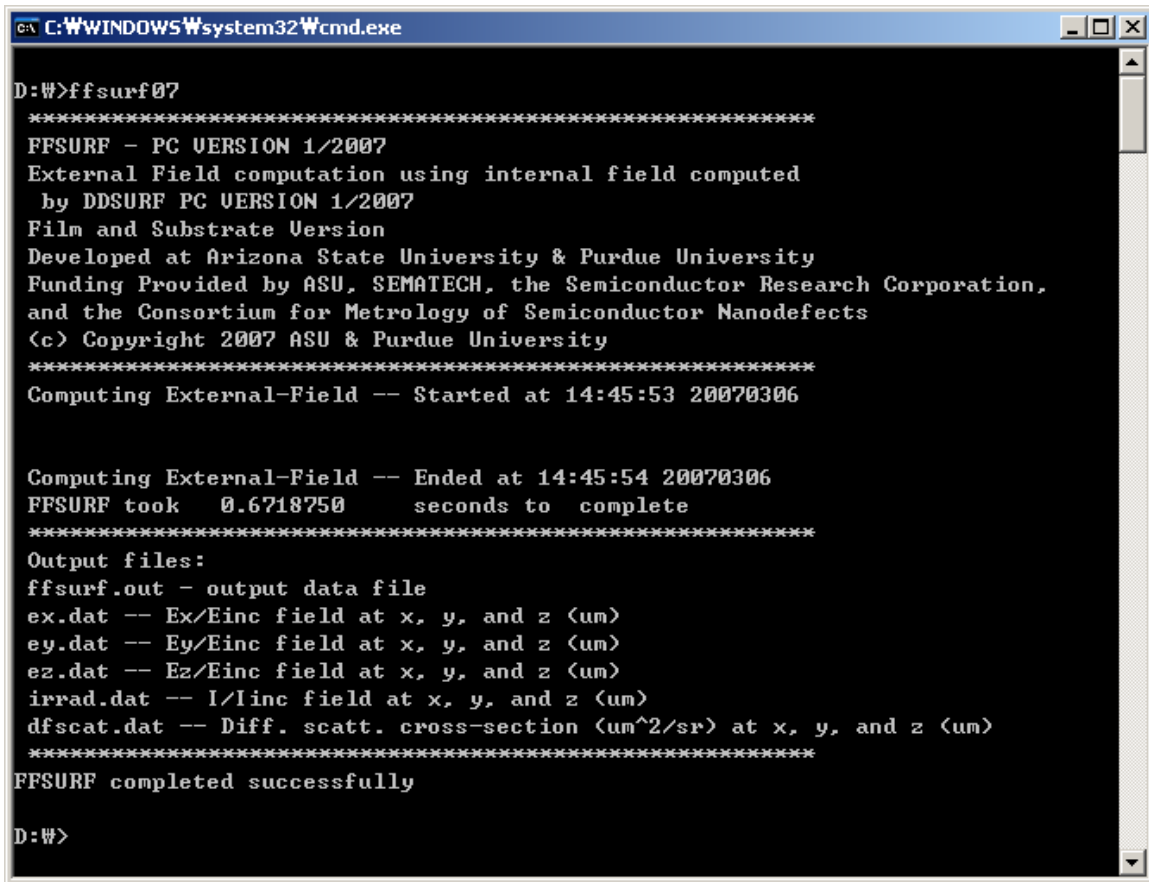
##### C.3.1.1 FFSURF Execution

The following will determine the field distribution in the plane along the  $x$  direction, perpendicular to the plane of incidence of the incident beam.

FFSURF uses the output file of DDSURF, *ddsurf.out*, as the basis to determine the external scattered field. The following should be entered into the FFSURF interface to create a plane detection region perpendicular to the plane of incidence and  $250 \mu\text{m}$  from the feature. The detection region is  $500 \mu\text{m} \times 500 \mu\text{m}$  in size, and discrete points of detection are every  $50 \mu\text{m}$  in the  $x$ ,  $y$ , and  $z$  directions.



When the "Execute FFSURF" is pressed after the parameters have been set, the screen output shows the time taken for computation as well as states the output files which were created:



### C.3.1.2 FFSURF output files – Plane detection region

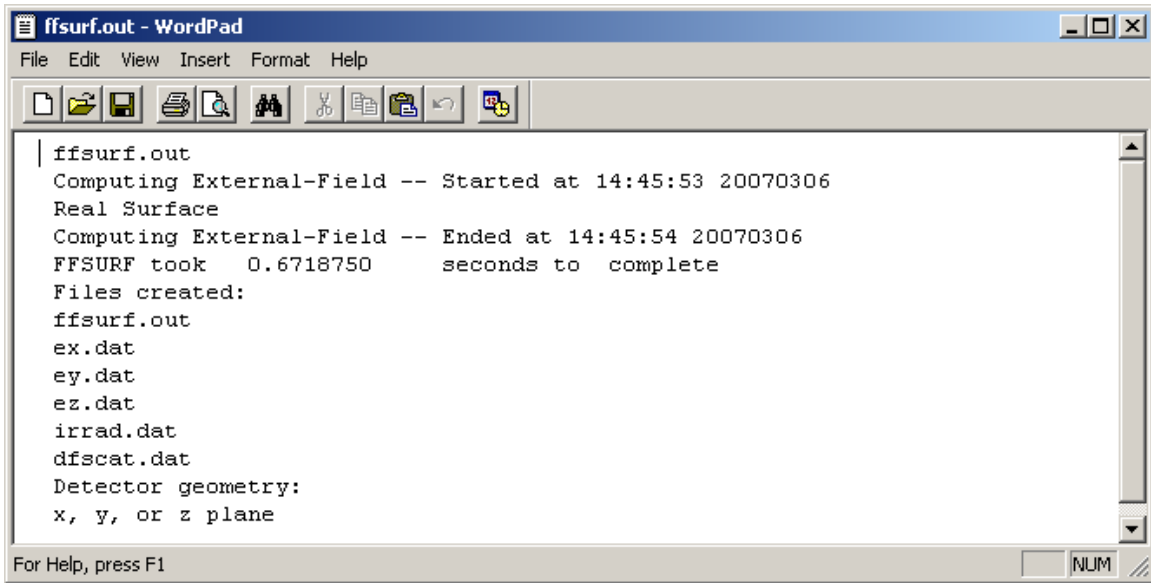
In this section, we will describe the output files which are created with the execution of FFSURF. These files will include:

- ffsurf.out – output data file for the case ran by FFSURF
- ex.dat - The x component of the electric field at the points of detection.
- ey.dat - The y component of the electric field at the points of detection
- ez.dat – The z component of the electric field at the points of detection

irrad.dat - The irradiance distribution at the points of detection based on the x, y, and z positions of the detection points. Irradiance is normalized by the incident field irradiance (1).  
dfscat.dat - Differential scattering cross section at points of detection.

### **ffsurf.out**

The following shows the *ffsurf.out* file which shows information about the run that was just completed. This is performed as a check for the user to make sure the correct set of parameters was used by FFSURF. When total cross-section is computed, it is stated in this file:



```
ffsurf.out
Computing External-Field -- Started at 14:45:53 20070306
Real Surface
Computing External-Field -- Ended at 14:45:54 20070306
FFSURF took 0.6718750 seconds to complete
Files created:
ffsurf.out
ex.dat
ey.dat
ez.dat
irrad.dat
dfscat.dat
Detector geometry:
x, y, or z plane
```

### **Electric Fields**

First we will look at an example of the external electromagnetic field which displays the complex E field at the points of detection. The electromagnetic fields which are given in terms of x, y, and z coordinates are  
ex.dat - The x component of the electric field at the points of detection.  
ey.dat - The y component of the electric field at the points of detection  
ez.dat - The z component of the electric field at the points of detection

An example of the **ex.dat** is shown in the following. The first three columns are the x, y, and z points of detection in microns. The fourth column is the real component of the electric field normalized to the incident field, and the fifth column is the imaginary component of the electric field normalized to the incident field.

```

ex.dat - WordPad
File Edit View Insert Format Help
| ex.dat
x (um)      y (um)      z (um)      Ex/Eo (real)  Ex/Eo (imag)
-0.2500000E+03 -0.2500000E+03 0.0000000E+00 -0.3004704E-06 0.9577546E-07
-0.2500000E+03 -0.2000000E+03 0.0000000E+00 -0.1875061E-06 -0.3413993E-06
-0.2500000E+03 -0.1500000E+03 0.0000000E+00 -0.4198874E-06 0.1365262E-06
-0.2500000E+03 -0.1000000E+03 0.0000000E+00 0.1137538E-06 0.4035598E-06
-0.2500000E+03 -0.5000000E+02 0.0000000E+00 -0.2467061E-06 -0.1046760E-06
-0.2500000E+03 0.0000000E+00 0.0000000E+00 -0.3253755E-09 -0.8924995E-09
-0.2500000E+03 0.5000000E+02 0.0000000E+00 0.2594608E-06 -0.6108432E-07
-0.2500000E+03 0.1000000E+03 0.0000000E+00 -0.4167441E-06 -0.4104960E-07
-0.2500000E+03 0.1500000E+03 0.0000000E+00 -0.1732779E-06 -0.4062837E-06
-0.2500000E+03 0.2000000E+03 0.0000000E+00 0.2279059E-06 -0.3164241E-06
-0.2500000E+03 0.2500000E+03 0.0000000E+00 -0.2689609E-06 -0.1655938E-06
-0.2500000E+03 -0.2500000E+03 0.5000000E+02 -0.1419588E-03 -0.5526869E-04
-0.2500000E+03 -0.2000000E+03 0.5000000E+02 -0.1057206E-03 0.1482722E-03
-0.2500000E+03 -0.1500000E+03 0.5000000E+02 -0.3839829E-04 -0.1963491E-03
-0.2500000E+03 -0.1000000E+03 0.5000000E+02 0.1832138E-03 0.2463162E-04
-0.2500000E+03 -0.5000000E+02 0.5000000E+02 0.2071661E-04 -0.1140180E-03
-0.2500000E+03 0.0000000E+00 0.5000000E+02 0.3801980E-06 0.1456689E-06
-0.2500000E+03 0.5000000E+02 0.5000000E+02 0.4919383E-04 0.1042276E-03
-0.2500000E+03 0.1000000E+03 0.5000000E+02 -0.9278514E-04 0.1595892E-03
-0.2500000E+03 0.1500000E+03 0.5000000E+02 0.1934771E-03 -0.5107420E-04
-0.2500000E+03 0.2000000E+03 0.5000000E+02 -0.1786765E-03 -0.3604689E-04
-0.2500000E+03 0.2500000E+03 0.5000000E+02 -0.4750714E-04 -0.1449521E-03
-0.2500000E+03 -0.2500000E+03 0.1000000E+03 0.2130792E-03 -0.1470425E-03
-0.2500000E+03 -0.2000000E+03 0.1000000E+03 -0.1019210E-03 0.2819714E-03
For Help, press F1
NUM

```

The other files, Ey and Ez are of the same form as shown before. When a radial geometry of detection is specified, Ex\_r.dat, Ey\_r.dat, and Ez\_r.dat are created, where the detector positions are given in terms of theta, phi, and radius.

### Irradiance Fields

Next, we look at the irradiance fields which were computed from the electromagnetic fields. The two irradiance files which can be created are:

irrad.dat - The irradiance distribution at the points of detection based on the x, y, and z positions of the detection points. Irradiance is normalized by the incident field irradiance (1).

Irrad\_r.dat - When a radial detection region is designated, this is the irradiance distribution at the points of detection based on theta, phi, and radius from the system origin. Irradiance is normalized by the incident field irradiance (1).

An example portion of the **irrad.dat** file is as follows:

irrad.dat - WordPad

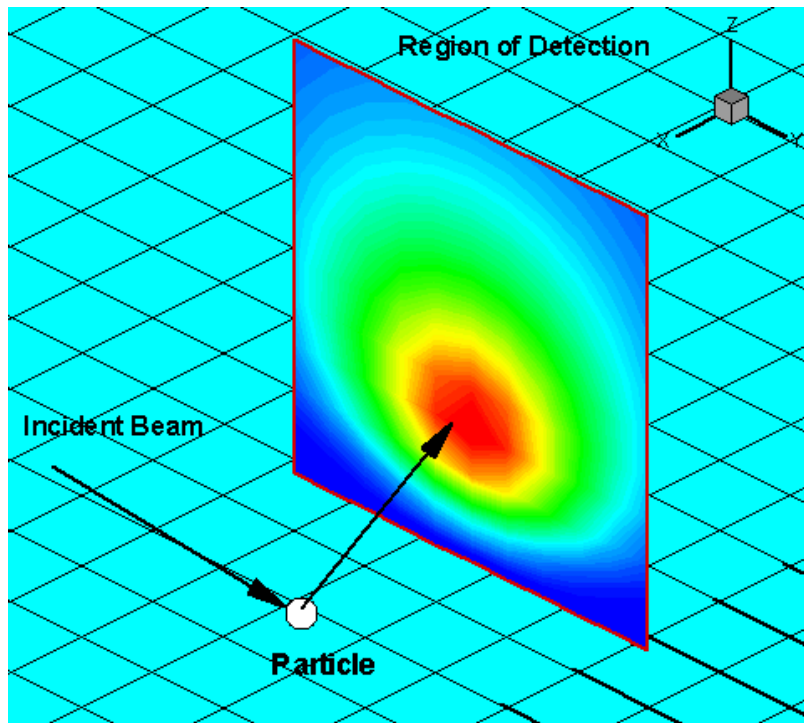
File Edit View Insert Format Help

irrad.dat

x (um)	y (um)	z (um)	I/ I <sub>0</sub>
-0.250000E+03	-0.250000E+03	0.000000E+00	0.2137903E-12
-0.250000E+03	-0.200000E+03	0.000000E+00	0.2778368E-12
-0.250000E+03	-0.150000E+03	0.000000E+00	0.3195328E-12
-0.250000E+03	-0.100000E+03	0.000000E+00	0.2970249E-12
-0.250000E+03	-0.500000E+02	0.000000E+00	0.2097947E-12
-0.250000E+03	0.000000E+00	0.000000E+00	0.1546541E-12
-0.250000E+03	0.500000E+02	0.000000E+00	0.2091856E-12
-0.250000E+03	0.100000E+03	0.000000E+00	0.2966120E-12
-0.250000E+03	0.150000E+03	0.000000E+00	0.3196367E-12
-0.250000E+03	0.200000E+03	0.000000E+00	0.2782226E-12
-0.250000E+03	0.250000E+03	0.000000E+00	0.2142065E-12
-0.250000E+03	-0.250000E+03	0.500000E+02	0.5673456E-07
-0.250000E+03	-0.200000E+03	0.500000E+02	0.9792028E-07
-0.250000E+03	-0.150000E+03	0.500000E+02	0.1667702E-06
-0.250000E+03	-0.100000E+03	0.500000E+02	0.2654618E-06
-0.250000E+03	-0.500000E+02	0.500000E+02	0.3674859E-06
-0.250000E+03	0.000000E+00	0.500000E+02	0.4136772E-06
-0.250000E+03	0.500000E+02	0.500000E+02	0.3680461E-06
-0.250000E+03	0.100000E+03	0.500000E+02	0.2661665E-06

For Help, press F1

The *irrad.dat* file is especially helpful to the user to visualize the strength of scattering from the scattering feature. The following shows an example of the irradiance distribution for sample case considered. The plot was created using TECPLOT, using the irradiance component as the contour plot, and the positions of detection. We can see that the strongest scattering (the color in the center) is nearly at specular scattering which is to be expected for this case.



### Differential Scattering Cross-Section

Another parameter of great interest which can be computed by the electromagnetic field which was determined by FFSURF is the differential scattering cross-section ( $dC_{sc}/d\Omega$ ). This is one of the most common parameters which are used to compare experimental scattering results to numerical scattering results. The differential scattering cross-sections are determined at the points of detection and are put in two data files:

dfscat.dat - The differential scattering cross-section distribution at the points of detection based on the x, y, and z positions of the detection points. The units for the differential scattering cross-sections are  $\mu\text{m}^2/\text{sr}$ .

dfscat\_r.dat - When a radial detection region is designated, this is the differential scattering cross-section at the points of detection based on theta, phi, and radius from the system origin. The units for the differential scattering cross-sections are  $\mu\text{m}^2/\text{sr}$ .

The following shows an example of the differential scattering cross-section file, *dfscat.dat* along the plane of detection.

```

dfscat.dat
x (um)      y (um)      z (um)      dCsa/dW (um^2/sr)
-0.250000E+03  -0.250000E+03  0.000000E+00  0.2672379E-07
-0.250000E+03  -0.200000E+03  0.000000E+00  0.2847828E-07
-0.250000E+03  -0.150000E+03  0.000000E+00  0.2716029E-07
-0.250000E+03  -0.100000E+03  0.000000E+00  0.2153430E-07
-0.250000E+03  -0.500000E+02  0.000000E+00  0.1363665E-07
-0.250000E+03  0.000000E+00  0.000000E+00  0.9665881E-08
-0.250000E+03  0.500000E+02  0.000000E+00  0.1359706E-07
-0.250000E+03  0.100000E+03  0.000000E+00  0.2150437E-07
-0.250000E+03  0.150000E+03  0.000000E+00  0.2716912E-07
-0.250000E+03  0.200000E+03  0.000000E+00  0.2851781E-07
-0.250000E+03  0.250000E+03  0.000000E+00  0.2677581E-07
-0.250000E+03  -0.250000E+03  0.500000E+02  0.7233657E-02
-0.250000E+03  -0.200000E+03  0.500000E+02  0.1028163E-01
-0.250000E+03  -0.150000E+03  0.500000E+02  0.1459239E-01
-0.250000E+03  -0.100000E+03  0.500000E+02  0.1990964E-01
-0.250000E+03  -0.500000E+02  0.500000E+02  0.2480529E-01
-0.250000E+03  0.000000E+00  0.500000E+02  0.2688902E-01
-0.250000E+03  0.500000E+02  0.500000E+02  0.2484311E-01
-0.250000E+03  0.100000E+03  0.500000E+02  0.1996249E-01

```

For Help, press F1

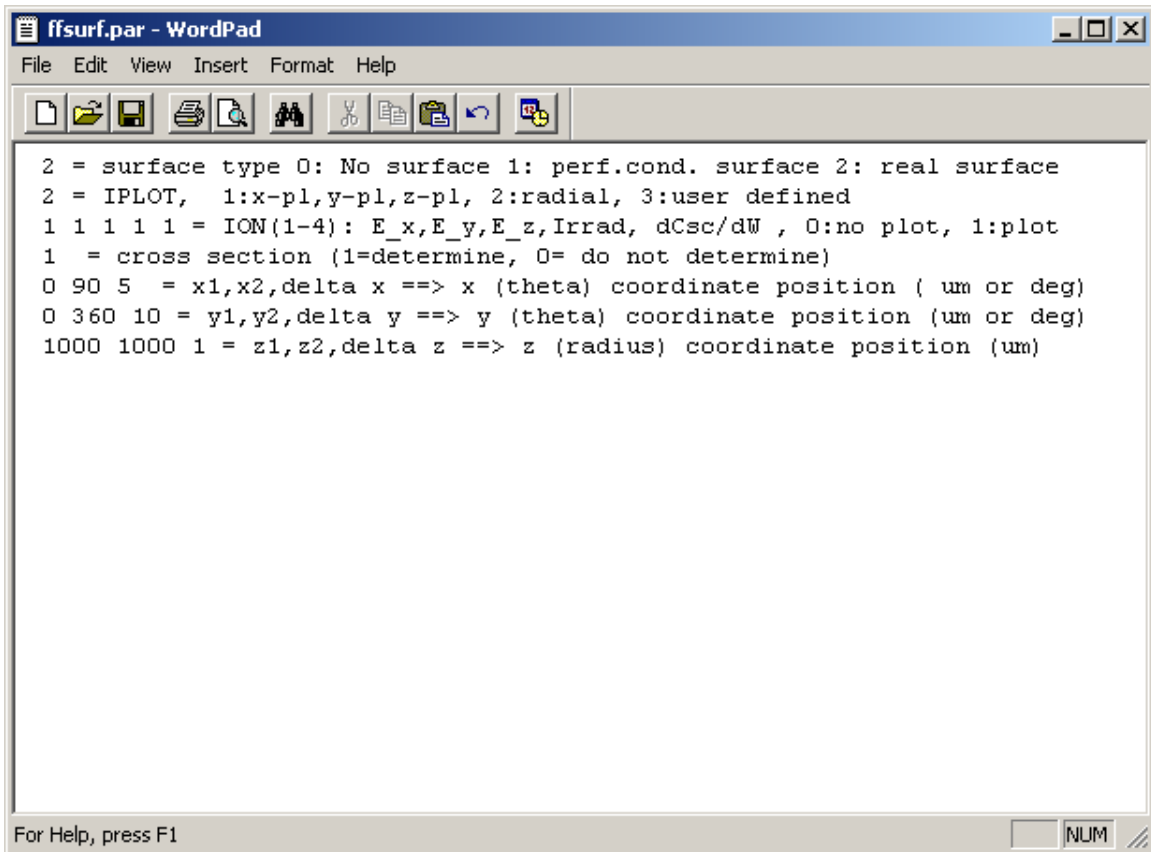


### C.3.2 Field Distribution along a hemispherical region of constant radius

Next, we will use the same feature configuration, but change the detection region. We now will determine the scattering field at a constant-radius hemispherical region. Since the region of interest is hemispherical, it is possible to determine the total scattering cross-section for the scattering feature above a surface.

#### C.3.2.1 FFSURF Execution

FFSURF uses the output file of DDSURF, *ddsurf.out*, as the basis to determine the external scattered field. The following shows how the interface should be set to create a hemispherical detection region at a constant radius of 1000  $\mu\text{m}$  about the (0,0,0) coordinates in terms of the dipole coordinate system. The scattered quantities are determined from  $0^\circ$  to  $90^\circ$  at  $5^\circ$  intervals for the  $\theta$  direction, and from  $0^\circ$  to  $360^\circ$  in  $10^\circ$  intervals in the  $\phi$  direction.



```
2 = surface type 0: No surface 1: perf.cond. surface 2: real surface
2 = IPLOT, 1:x-pl,y-pl,z-pl, 2:radial, 3:user defined
1 1 1 1 1 = ION(1-4): E_x,E_y,E_z,Irrad, dCsc/dW , 0:no plot, 1:plot
1 = cross section (1=determine, 0= do not determine)
0 90 5 = x1,x2,delta x ==> x (theta) coordinate position ( um or deg)
0 360 10 = y1,y2,delta y ==> y (theta) coordinate position (um or deg)
1000 1000 1 = z1,z2,delta z ==> z (radius) coordinate position (um)
```

When the "Execute FFSURF" is pressed after the parameters have been set, the screen output looks as the following:

```

C:\WINDOWS\system32\cmd.exe
D:\W>ffsurf07
*****
FFSURF - PC VERSION 1/2007
External Field computation using internal field computed
  by DDSURF PC VERSION 1/2007
Film and Substrate Version
Developed at Arizona State University & Purdue University
Funding Provided by ASU, SEMATECH, the Semiconductor Research Corporation,
and the Consortium for Metrology of Semiconductor Nanodefects
(c) Copyright 2007 ASU & Purdue University
*****
Computing External-Field -- Started at 14:53:58 20070306

Total Scattering Cross-Section (um^2) = 0.1549872
Computing External-Field -- Ended at 14:54:02 20070306
FFSURF took 3.937500 seconds to complete
*****
Output files:
ffsurf.out - output data file, total scattering cross-section (um^2)
ex.dat -- Ex/Einc field at x, y, and z (um)
ey.dat -- Ey/Einc field at x, y, and z (um)
ez.dat -- Ez/Einc field at x, y, and z (um)
irrad.dat -- I/linc field at x, y, and z (um)
ex_r.dat -- Ex/Einc field at theta, phi, and r (deg,um)
ey_r.dat -- Ey/Einc field at theta, phi, and r (deg,um)
ez_r.dat -- Ez/Einc field at theta, phi, and r (deg,um)
dfscat.dat -- Diff. scatt. cross-section (um^2/sr) at x, y, and z (um)
dfscat_r.dat -- Diff. scatt. cross-section
  (um^2/sr) at theta, phi, and radius (deg, um)
irrad_r.dat -- Irradiance field at theta, phi and radius (deg,um)
*****
FFSURF completed successfully

D:\W>

```

As in the previous examples, the time used to compute the scattered field and the output data files are mentioned in the data files.

### C.3.2.2 FFSURF Output Files – Hemispherical Detection Region

In this section, we will describe the output files which are created with the execution of FFSURF. From the screen output, we see the only difference than with the plane detection region case is the introduction to the *\_r* files, which are the same as the other data files, though the positions are given in theta, phi, and radius instead of Cartesian coordinates.

#### **ffsurf.out**

The following shows the *ffsurf.out* file which shows information about the run that was just completed. This is performed as a check for the user to make sure the correct set of parameters was used by FFSURF. When total cross-section is computed, it is stated in this file:

```
ffsurf.out
Computing External-Field -- Started at 14:53:58 20070306
Real Surface
Computing External-Field -- Ended at 14:54:02 20070306
FFSURF took 3.937500 seconds to complete
Files created:
ffsurf.out
ex.dat
ey.dat
ez.dat
ex_r.dat
ey_r.dat
ez_r.dat
irrad_r.dat
irrad.dat
dfscat.dat
dfscat_r.dat
Detector geometry:
Constant radius region
Scattering Cross-Section = 0.1549872 um^2
```

### Electric Fields

First we will look at an example of the external electromagnetic field which displays the complex E field at the points of detection. As in the case where the region of detection was a plane, when a radial detection region is designated, the electromagnetic fields are given in data files in terms of:

- ex.dat - The x component of the electric field at the points of detection.
- ey.dat - The y component of the electric field at the points of detection
- ez.dat - The z component of the electric field at the points of detection

These files give the x, y, and z components of the complex electric fields at detection points in terms of the Cartesian coordinate system. Now that the detection region is radial, three more electric field files are produced, called:

- ex\_r.dat - The x component of the electric field at the points of detection designated by theta, phi and radius.
- ey\_r.dat - The y component of the electric field at the points of detection designated by theta, phi and radius.
- ez\_r.dat - The z component of the electric field at the points of detection designated by theta, phi and radius.

An example of the **ex\_r.dat** is shown in the following. The first three columns are the theta, phi and radius for the points of detection in microns. The fourth column is the real component of the electric field normalized to the incident field, and the fifth column is the imaginary component of the electric field normalized to the incident field.

```

ex_r.dat
theta (deg)  phi (deg)  radius (um)  Ex/Eo(real)  Ex/Eo(imag)
0.000000E+00  0.000000E+00  0.100000E+04  -0.5368019E-09  0.7155383E-08
0.500000E+01  0.000000E+00  0.100000E+04  0.6838511E-09  0.4324295E-08
0.100000E+02  0.000000E+00  0.100000E+04  -0.7963623E-09  0.8510124E-09
0.150000E+02  0.000000E+00  0.100000E+04  -0.5371442E-09  0.6851062E-09
0.200000E+02  0.000000E+00  0.100000E+04  -0.1713083E-08  -0.2071759E-08
0.250000E+02  0.000000E+00  0.100000E+04  -0.1948221E-08  -0.2901440E-08
0.300000E+02  0.000000E+00  0.100000E+04  -0.2732681E-08  -0.4475922E-08
0.350000E+02  0.000000E+00  0.100000E+04  -0.3282126E-08  -0.4357233E-08
0.400000E+02  0.000000E+00  0.100000E+04  -0.3173879E-08  -0.4119089E-08
0.450000E+02  0.000000E+00  0.100000E+04  -0.4067399E-08  -0.6574315E-08
0.500000E+02  0.000000E+00  0.100000E+04  -0.3115669E-08  -0.5336172E-08
0.550000E+02  0.000000E+00  0.100000E+04  -0.2316331E-08  -0.6144900E-08
0.600000E+02  0.000000E+00  0.100000E+04  -0.1633510E-08  -0.6460186E-08
0.650000E+02  0.000000E+00  0.100000E+04  -0.1486823E-08  -0.5642300E-08
0.700000E+02  0.000000E+00  0.100000E+04  -0.9537873E-09  -0.5649850E-08
0.750000E+02  0.000000E+00  0.100000E+04  -0.9399550E-09  -0.5019136E-08
0.800000E+02  0.000000E+00  0.100000E+04  -0.5243070E-09  -0.3783129E-08
0.850000E+02  0.000000E+00  0.100000E+04  0.1951558E-12  -0.2109983E-08
0.900000E+02  0.000000E+00  0.100000E+04  0.2620491E-11  0.2374260E-11
0.000000E+00  0.100000E+02  0.100000E+04  -0.5368019E-09  0.7155383E-08

```

### Irradiance Fields

Next, we look at the irradiance fields which were computed from the electromagnetic fields. The two irradiance files which can be created are:

**irrad.dat** - The irradiance distribution at the points of detection based on the x, y, and z positions of the detection points. Irradiance is normalized by the incident field irradiance (1).

**Irrad\_r.dat** - When a radial detection region is designated, this is the irradiance distribution at the points of detection based on theta, phi, and radius from the system origin. Irradiance is normalized by the incident field irradiance (1).

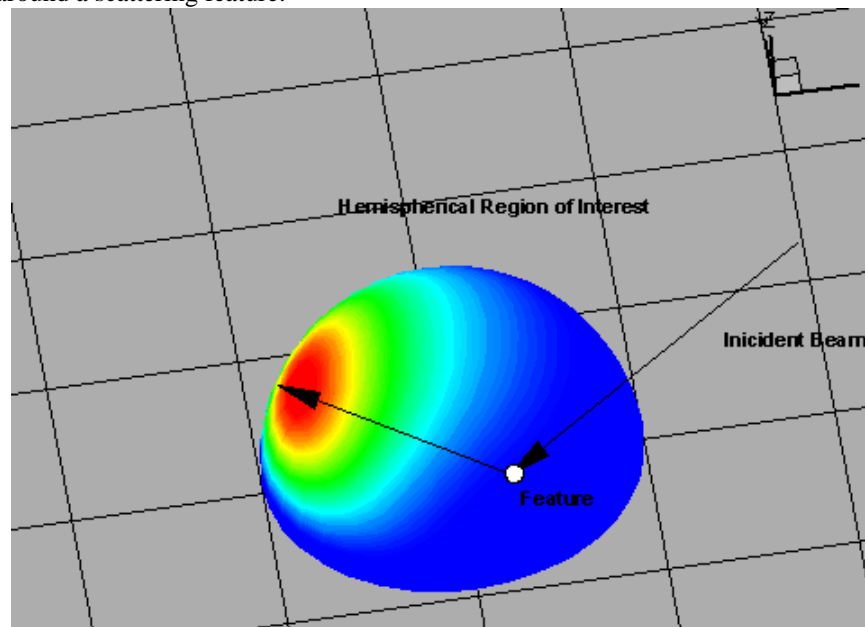
The irradiance distribution can also be given in terms of  $\theta$ ,  $\phi$  and radius. **irrad\_r.dat** has the following form:

```

irrad_r.dat
theta (deg)  phi (deg)  radius (um)  I/Io
0.000000E+00  0.000000E+00  0.100000E+04  0.3689630E-07
0.500000E+01  0.000000E+00  0.100000E+04  0.3140133E-07
0.100000E+02  0.000000E+00  0.100000E+04  0.2664193E-07
0.150000E+02  0.000000E+00  0.100000E+04  0.2246981E-07
0.200000E+02  0.000000E+00  0.100000E+04  0.1877529E-07
0.250000E+02  0.000000E+00  0.100000E+04  0.1549475E-07
0.300000E+02  0.000000E+00  0.100000E+04  0.1259972E-07
0.350000E+02  0.000000E+00  0.100000E+04  0.1009767E-07
0.400000E+02  0.000000E+00  0.100000E+04  0.8004921E-08
0.450000E+02  0.000000E+00  0.100000E+04  0.6332588E-08

```

The irradiance distribution is of particular interest to the users, as comparisons can be made with experiments, as well as a quick computation can give the user a feel for the pattern of scattering that results from the scattering system. The following shows a hemispherical representation of the irradiance distribution around a scattering feature.



**Differential Scattering Cross-Section**

Another parameter of great interest which can be computed by the electromagnetic field which was determined by FFSURF is the differential scattering cross-section ( $dC_{sc}/d\Omega$ ). This is one of the most common parameters which are used to compare experimental scattering results to numerical scattering results. The differential scattering cross-sections are determined at the points of detection and are put in two data files:

dfscat.dat - The differential scattering cross-section distribution at the points of detection based on the x, y, and z positions of the detection points. The units for the differential scattering cross-sections are  $\mu\text{m}^2/\text{sr}$ .

dfscat\_r.dat - When a radial detection region is designated, this is the differential scattering cross-section at the points of detection based on theta, phi, and radius from the system origin. The units for the differential scattering cross-sections are  $\mu\text{m}^2/\text{sr}$ .

The following shows an example of the differential scattering cross-section file, *dfscat\_r.dat* for the hemispherical region of detection:

dfscat\_r.dat - WordPad

File Edit View Insert Format Help

dfscat\_r.dat

theta (deg)	phi (deg)	radius(um)	dCsa/dW (um <sup>2</sup> /sr)
0.0000000E+00	0.0000000E+00	0.1000000E+04	0.3689630E-01
0.5000000E+01	0.0000000E+00	0.1000000E+04	0.3140134E-01
0.1000000E+02	0.0000000E+00	0.1000000E+04	0.2664193E-01
0.1500000E+02	0.0000000E+00	0.1000000E+04	0.2246981E-01
0.2000000E+02	0.0000000E+00	0.1000000E+04	0.1877529E-01

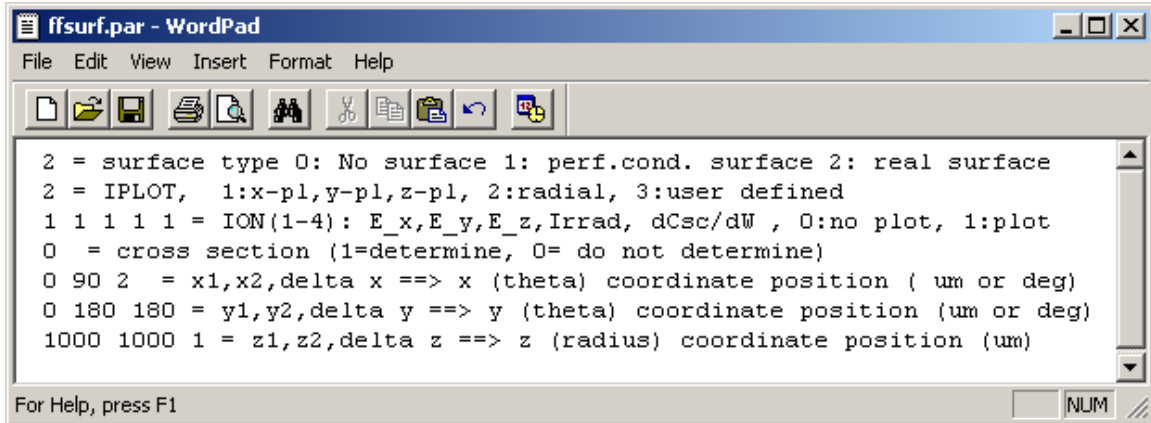
For Help, press F1

NUM

### C.3.3 Field Distribution along plane of incidence at constant radius

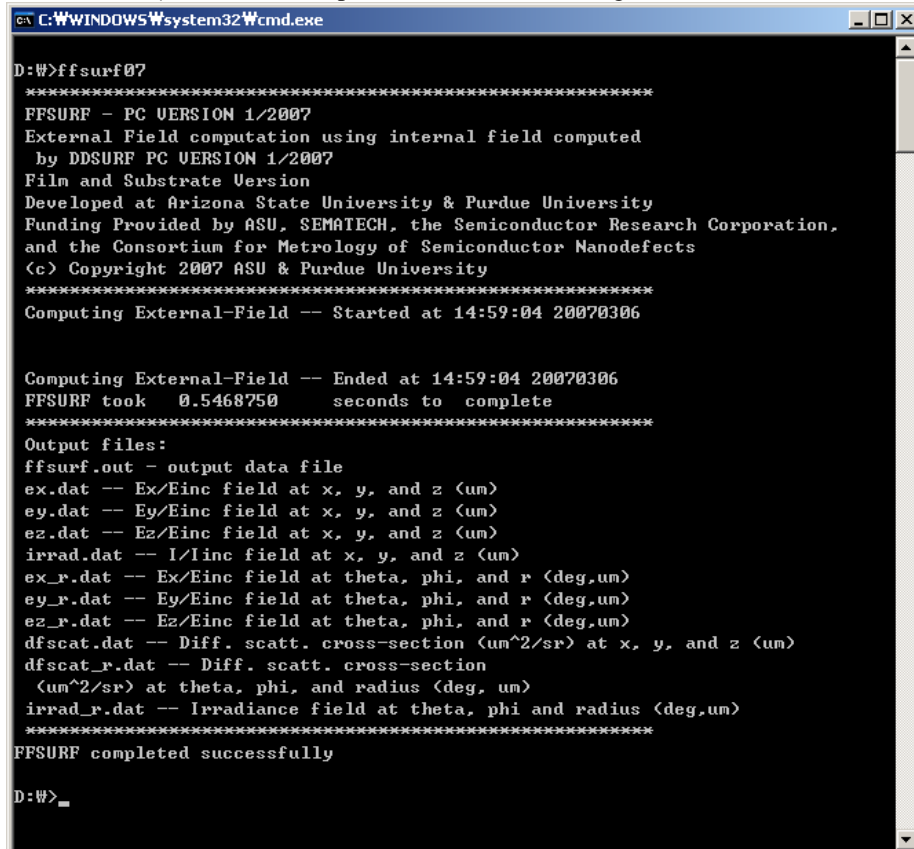
#### C.3.3.1 FFSURF Execution

Another interesting configuration that can be investigated is the irradiance distribution along the plane of incidence, and at a constant radius from the feature. This is useful, as the number of discrete points computed is not as numerous as an hemisphere region, thus reducing the time required to run FFSURF to get meaningful results. The following shows an example set of input parameters which can be used by FFSURF to determine the field along a plane of incidence. We shall have the following input file to find irradiance along the plane of incidence at a distance of 1000  $\mu\text{m}$  from the spherical feature:



```
ffsurf.par - WordPad
File Edit View Insert Format Help
2 = surface type 0: No surface 1: perf.cond. surface 2: real surface
2 = IPLOT, 1:x-pl,y-pl,z-pl, 2:radial, 3:user defined
1 1 1 1 1 = ION(1-4): E_x,E_y,E_z,Irrad, dCsc/dW , 0:no plot, 1:plot
0 = cross section (1=determine, 0= do not determine)
0 90 2 = x1,x2,delta x ==> x (theta) coordinate position (um or deg)
0 180 180 = y1,y2,delta y ==> y (theta) coordinate position (um or deg)
1000 1000 1 = z1,z2,delta z ==> z (radius) coordinate position (um)
For Help, press F1 NUM
```

For this input, detection points will be computed from 0° to 90° at 2° intervals for the  $\theta$  direction and along the 0° and 180° lines of  $\phi$ . The screen output will have the following form:



```
C:\WINDOWS\system32\cmd.exe
D:\>ffsurf07
*****
FFSURF - PC VERSION 1/2007
External Field computation using internal field computed
by DDSURF PC VERSION 1/2007
Film and Substrate Version
Developed at Arizona State University & Purdue University
Funding Provided by ASU, SEMATECH, the Semiconductor Research Corporation,
and the Consortium for Metrology of Semiconductor Nanodefects
(c) Copyright 2007 ASU & Purdue University
*****
Computing External-Field -- Started at 14:59:04 20070306

Computing External-Field -- Ended at 14:59:04 20070306
FFSURF took 0.5468750 seconds to complete
*****
Output files:
ffsurf.out - output data file
ex.dat -- Ex/Einc field at x, y, and z (um)
ey.dat -- Ey/Einc field at x, y, and z (um)
ez.dat -- Ez/Einc field at x, y, and z (um)
irrads.dat -- I/line field at x, y, and z (um)
ex_r.dat -- Ex/Einc field at theta, phi, and r (deg,um)
ey_r.dat -- Ey/Einc field at theta, phi, and r (deg,um)
ez_r.dat -- Ez/Einc field at theta, phi, and r (deg,um)
dfscat.dat -- Diff. scatt. cross-section (um^2/sr) at x, y, and z (um)
dfscat_r.dat -- Diff. scatt. cross-section
(um^2/sr) at theta, phi, and radius (deg, um)
irrads_r.dat -- Irradiance field at theta, phi and radius (deg,um)
*****
FFSURF completed successfully
D:\>_
```

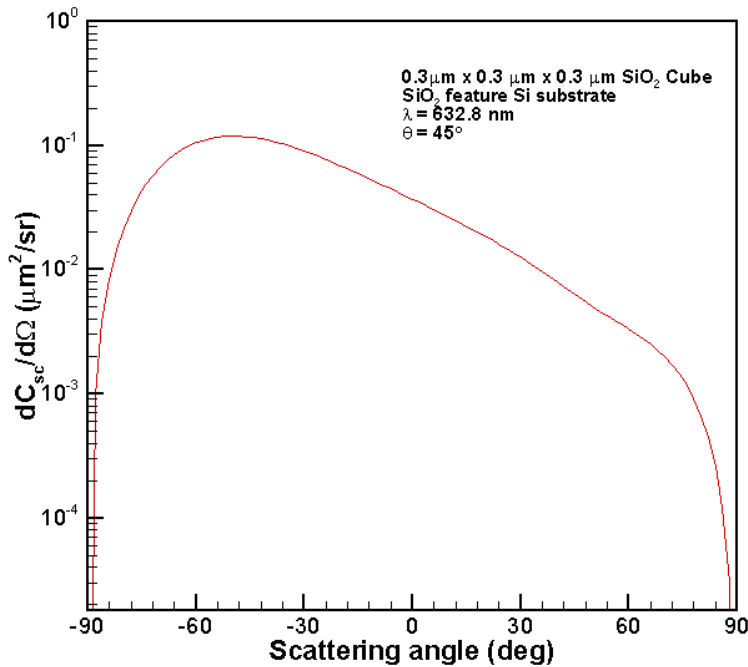
As in the previous examples, the time used to compute the scattered field and the output data files are mentioned in the data files.

### C.3.3.2 FFSURF Output Files – Plane of Incidence – Constant Radius

Since this is a subset of the previous example of a hemispherical detection region, we will just show an example which is useful when comparing to experimental results, that is, the differential scattering cross-section. Actual comparisons with this have been shown in Chapter 3 of this manual. To plot the differential scattering cross-section, we will plot the radial form of the differential scattering cross-section, that is, *dfscat\_r.dat*:

theta (deg)	phi (deg)	radius (um)	dCsa/dW (um <sup>2</sup> /sr)
0.000000E+00	0.000000E+00	0.100000E+04	0.3689630E-01
0.200000E+01	0.000000E+00	0.100000E+04	0.3460190E-01
0.400000E+01	0.000000E+00	0.100000E+04	0.3243645E-01
0.600000E+01	0.000000E+00	0.100000E+04	0.3039663E-01
0.800000E+01	0.000000E+00	0.100000E+04	0.2846914E-01
0.100000E+02	0.000000E+00	0.100000E+04	0.2664193E-01

When the differential scattering cross-sections are plotted, we get a simple plot which can help us in our scattering evaluation:



Differential Scattering Cross-section distribution at a constant radius along the plane of beam incidence.