

Copyright © 1995, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**TEMPEST USERS' GUIDE**

**Version 4.0**

by

Alfred Wong

Memorandum No. UCB/ERL M95/14

14 March 1995

**TEMPEST USERS' GUIDE**

**Version 4.0**

by

**Alfred Wong**

**Memorandum No. UCB/ERL M95/14**

**14 March 1995**

**ELECTRONICS RESEARCH LABORATORY**

**College of Engineering  
University of California, Berkeley  
94720**

**Table of Contents**

---

Copyright Notice and Limitation of Liability ..... iii

Contributors ..... viii

1.0 Introduction ..... 1

    1.1 Overview ..... 1

    1.2 New in Version 4.0 ..... 2

    1.3 Applications Information ..... 2

    1.4 Source Code Information ..... 3

    1.5 About this Manual ..... 4

2.0 Installation ..... 5

    2.1 Resource Requirements ..... 5

    2.2 Installation ..... 5

    2.3 Compilation ..... 6

    2.4 Testing ..... 7

3.0 Tutorial ..... 8

    3.1 Overview ..... 8

    3.2 Input File to TEMPEST ..... 8

    3.3 Checking the Input File ..... 11

    3.4 Running Simulations ..... 11

    3.5 Output Data ..... 13

    3.6 Example 2: Photoresist Exposure ..... 18

4.0 Detail Command Descriptions ..... 24

    4.1 Input File Keywords ..... 24

    4.2 TEMPEST commands ..... 32

    4.3 Post-TEMPEST Workstation Commands ..... 32

---

**Table of Contents**

---

5.0 Useful Hints .....36

    5.1 Top and Bottom of the Simulated Structure.....36

    5.2 Material Interface from the Top and Bottom Boundaries .....36

    5.3 Verification of Loaded Topography.....36

    5.4 Real and Imaginary Parts of the Refractive Index .....36

    5.5 Number of Simulation Nodes.....36

    5.6 Periodic and Isolated Structures .....37

    5.7 Convergence Problems.....37

References .....38

Summary of Commands.....39

---

**Copyright Notice and Limitation of Liability**

---

Copyright (c) 1995. The Regents of the University of California. All rights reserved.

SAMPLE-3D version 2.0 was written principally by Kenny K.H. Tbh, Ed Scheckler, John Helmsen, and John Sefer with assistance from Calvin Cheng, and Ravi P. Gunturi under the direction of A.R. Neureuther in the SAMPLE Group in the Electrical Engineering and Computer Science Department of the University of California, Berkeley through support in part by SRC/Sematech.

SAMPLE -3D is not public domain software. The use and redistributions are subject to the ILP Research Software License Agreement as amended by the additional restrictions on SAMPLE GROUP software.

---

**Research Software License Agreement**

---

(revised October 1994)

This form must be signed and enclosed with your order of any software or documentation.

This form specifies the terms under which the software and documentation referenced in this booklet are provided. Some specified software and documentation are subject to special licensing terms which are described within. Those terms are incorporated into this agreement.

Software is distributed as is, completely without warranty or service support. The University of California and its employees are not liable for the condition or performance of the software.

The University owns the copyright in and all other proprietary rights to all software and documentation provided under this agreement.

Notwithstanding the foregoing, the University does not warrant that it owns such copyright or other proprietary rights. The University shall not be liable for any infringement of copyright or other proprietary rights brought by third parties against the licensee of the software and documentation provided under this agreement.

**THE UNIVERSITY OF CALIFORNIA HEREBY DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE UNIVERSITY IS NOT LIABLE FOR ANY DAMAGES INCURRED BY THE RECIPIENT IN USE OF THE SOFTWARE AND DOCUMENTATION, INCLUDING DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.**

The University of California grants the licensee the right to modify, copy, and redistribute the software and documentation, both within the licensee's organization and externally, subject to the following restrictions:

The licensee agrees not to charge for the University of California code itself.

The licensee may, however, charge for additions, extensions, or support. In any product based on the software, the licensee agrees to acknowledge the research group that developed the software. This acknowledgment shall appear in the product documentation.

The licensee agrees to obey all U.S. Government restrictions governing redistribution or export of the software and documentation. For software with addi-

---

**Research Software License Agreement**

---

tional restrictions (any software listed in Section 2), the licensee agrees to all additional terms governing distribution of that software. The licensee agree to reproduce any copyright notice which appears on the software and documentation provided under this agreement on any copy or modification of such made available to others by licensee.

**SOFTWARE and/or DOCUMENTATION ORDERED:**

**SIGNATURE AND DATE:**

**NAME (please print or type):**

**TITLE:**

**ORGANIZATION:**



---

**Additional Restrictions on SAMPLE Group Software**

---

These additional restrictions of the Research Software License Agreement apply to software developed by students in the SAMPLE Group of Professor A. R. Neureuther in the Electrical Engineering and Computer Sciences Department of the University of California at Berkeley, including projects sponsored by Semiconductor Research Corporation / SEMATECH. This form must also be signed and enclosed with your order of any software and documentation from the SAMPLE Group.

TEMPEST 4.0 may not be exported without prior written permission of SRC/SEMATECH before July 1, 1997.

This software is subject to the specific conditions of SRC/SEMATECH which are,

“The contract specifies that worldwide, unrestricted, royalty-free, nonexclusive rights to research results be licensed to the SRC and that the SRC be authorized to license such rights to its members.”

The licensee may use this code and its derivative works royalty free in commercial products, work for hire, and university/institute software development projects provided that the University of California is formally notified immediately upon entering into contracts/grants to do so and again simultaneously with announcements of products or software tools.

The licensee agrees to not permit public access to the code or its derivative works and further agrees that in making any redistribution of the code or its derivative works to propagated all the terms of this agreement and to provide a mechanism to trace down any and every copy of the code resulting from their redistribution.

**SOFTWARE and/or DOCUMENTATION:**

**SIGNATURE AND DATE:**

**NAME (please print or type):**

**TITLE:**

**ORGANIZATION:**

---

**LETTER OF ASSURANCE**

---

---

**LETTER OF ASSURANCE**

---

**\*\*This is only a SAMPLE. Please use your company letterhead for your letter\*\***

May 8th, 1995

Industrial Liason Program  
University of California at Berkeley  
Berkeley, CA 94720-1770  
U.S.A.

To whom it may concern:

[Name of Organization or Institution] hereby assures the University of California that the software and documentation provided hereunder or any immediate product (including processes and services) produced directly by use of the software and/or documentation is not intended for, and will not be shipped, either directly or indirectly, to Iraq, Montenegro, Serbia, or any country listed in Country Group S (Libya), and Z (Cuba, North Korea, Vietnam), as specified in Part 770 of the Export Administration Regulations of the United States Department of Commerce, and the Foreign Assets Control Regulation of the United States Department of the Treasury.

[Name of Organization or Institution] hereby further assures the University of California that the software and documentation provided hereunder and the immediate product (including processes and services) produced directly by use of the software and/or documentation will not be made available to, or for use by, or for the military or police entities of the Republic of South Africa as specified in the Export Administration Regulations of the United States Department of Commerce.

The provisions of this letter of assurance shall survive and continue after any termination of any agreement under which the software and documentation are provided.

Sincerely,

Signed by member of your organization

[Title]

---

**Contributors**

---

This is the fourth release of TEMPEST. The previous release dates were:

ver1.0 July 23, 1991.  
ver2.0 December 31, 1992.  
ver3.0 July 14, 1994.

The following people have contributed to TEMPEST:

Original Authors:

John Gamelin(ver1.0)  
Roberto Guerrieri(ver1.0)  
Karim Tadros(ver1.0)  
Alfred Wong(ver2.0, 3.0, and 4.0)

Contributors to earlier releases:

Takeshi Doi(ver1.0)  
Eric Hall(Manual Format)

TEMPEST is developed at the University of California at Berkeley by a student research group on process modeling and technology with Professor Andrew R. Neureuther. The development of TEMPEST has been primarily supported by a SRC/SEMATECH grant, as well as computer time from the Thinking Machines Corporation and the National Center for Supercomputing Applications (NCSA).

An initial \*LISP version for the CM-2 architecture was written by John Gamelin and extended and tested by Karim Tadros. Alfred Wong converted the code to C\*, added TM polarization, off-axis illumination, and post process image synthesis in version 2.0. In version 3.0, he added three-dimensional analysis capabilities on the CM-5 and linked TEMPEST with SPLAT. Version 4.0 has similar capabilities with version 3.0, except that version 4.0 is written in C and can be executed on work stations and personal computers.

To encourage open exchange of information, TEMPEST and its supporting utilities are available on an as-is basis for a small handling fee. However, there is usually no assistance available for implementing the program or in training users. Although a prompt response cannot be promised, feedback on the program and comments are welcome. Please mail to:

Professor Andrew R. Neureuther  
EECS Department  
510 Cory Hall  
University of California, Berkeley  
Berkeley, CA 94720  
U. S. A.  
(510)642-4590  
neureuth@eecs.berkeley.edu

---

# TEMPEST version 4.0

---

**Alfred Wong**

**Electronics Research Laboratory**

University of California, Berkeley

---

## **1.0 Introduction**

---

### **1.1 Overview**

---

The acronym TEMPEST<sup>3</sup> stands for "Time-domain Electromagnetic Massively Parallel Evaluation of Scattering from Topography." The computer program solves Maxwell's equations using a time-domain finite-difference algorithm, where the electric and magnetic field nodes are spatially and temporally staggered over a three-dimensional topography of interest. Version 3.0 takes advantage of the inherent parallel nature of electromagnetic wave propagation and is implemented on the computer architecture connection machine 5 (CM-5)<sup>8</sup>. Due to the limited availability of the CM-5, version 4.0 is implemented on any single-processor computer architecture such as a work station or even a person computer. The simulation domain may represent periodic or symmetric topography. The algorithm is capable of simulating problems such as scattering from asymmetrical alignment marks, transmission through phase-shifting masks, effects of line-edge profiles in metrology as well as dynamic bleaching of photoresist over arbitrary non-planar, inhomogeneous wafer topographies.

Illumination is assumed to be monochromatic, with the electric field linearly polarized in any user-specified direction. The incident angle can take on discrete values depending on the illumination wavelength and the dimension of the simulation domain. Illumination is assumed to be coherent and can consist of any intensity profile such as that calculated from SPLAT<sup>9</sup>. The steady-state electric fields as well as the magnitudes and phases of diffraction harmonics are found. The photosensitizer concentration is calculated if a layer of photoresist is present. The matrix containing this concentration information can be used in

other simulation programs such as SAMPLE<sup>4</sup> for simulation of resist development. Information on the diffraction harmonics is used to form optical image profiles in SPLAT<sup>9</sup>. Imaging can be done on the scattered fields (for example, reflected light from alignment marks) or the transmitted fields (for example, light passing through a phase-shift mask).

TEMPEST parses topography information from an input file which can be checked for correctness. The input geometry is then simulated until the electromagnetic field reaches steady-state or, in the case of non-convergence, the simulation domain is excited for a user chosen number of wave cycles. Information on the simulation parameters as well as the diffraction harmonics are written to an output file. Optical image profiles based on the diffraction harmonics can be generated by the simulation program SPLAT<sup>9</sup>. Line plots, contour plots, and/or density plots of the field amplitude, steady-state field, transient field, refractive index and photo active compound (PAC) concentration in any region of the simulation domain can be generated using the plotting package PLOTMTV.

---

## **1.2 New in Version 4.0**

---

Version 4.0 of the program TEMPEST has several extensions from the previous version (version 3.0). Three-dimensional problems can be analyzed on a work station instead of a supercomputer. Furthermore, the number of simulation nodes in the height dimension (the z-direction) is no longer restricted to powers of 2. This represents a savings in the amount of memory needed. Checking of the input file is no longer done by the auxiliary routine CHECK. Instead, invoking TEMPEST with only the input file name checks the input file for correctness.

Besides the main electromagnetic fields solver program, TEMPEST version 4.0 includes several auxiliary programs. The programs AMPHA1D and AMPHA2D allow the user to calculate the amplitude and phase of any electromagnetic field component along any line or plane. TOTAL1D and TOTAL2D are used to calculate the total electric or magnetic field along a line and a plane respectively. REDUCE2D is used to reduce data volume and to ease output data processing.

---

## **1.3 Applications Information**

---

TEMPEST is capable of simulating a number of optical phenomena occurring in photolithography and optical metrology applications. These include bleaching of photoresist over arbitrary topography (e.g., reflective notching), dark-field and bright-field imaging of wafer features (e.g., alignment marks and line structures), generation of image profiles through masks (e.g., reduction phase-shift mask), and calculation of the electric and magnetic fields at all

points in the three-dimensional simulation domain at any instant in time. Several studies involving these topics have already been performed, and the user is encouraged to refer to these publications<sup>2,3,6,7,11,12,13,14</sup>. In general, TEMPEST simulations can be placed into two categories:

1. Lithography which involves dynamic changes during exposure, and
2. Imaging and scattering analysis.

These categories are distinguished primarily by the desired output. For lithography the desired output is the final PAC concentration. For imaging/scattering purposes the primary output desired are the electromagnetic fields and the diffraction harmonics. These data are used by the program SPLAT to form the bright-field or dark-field images. Details of the image synthesis procedure are described in Section 4.3.1. The other desirable output for an imaging/scattering analysis is the field values in the simulation domain. Line plots, contour plots, surface plots and/or density plots of the transient field and the steady-state field, the PAC concentration and the refractive index can be generated. Generation of these plots are described in Section 4.1.5.

---

## 1.4 Source Code Information

---

TEMPEST ver4.0 is written in the C programming language. The source code is distributed into twelve text files. Six header files are also included. The header files are:

<b>defn.h</b>	constant definitions
<b>global.h</b>	global variable definitions
<b>header.h</b>	header files to be included
<b>lib.h</b>	libraries to include
<b>routn.h</b>	subroutines used in the program
<b>ver.h</b>	version and release date information

The files which contain the source code are:

<b>bc.c</b>	boundary conditions
<b>bulk.c</b>	interior updating equations
<b>de.c</b>	computes the diffraction harmonics
<b>in.c</b>	reads information from the input file
<b>init.c</b>	initialization of variables
<b>iterate.c</b>	fields computation by iteration
<b>loop.c</b>	calls different loops according to the model
<b>main.c</b>	program control
<b>math.c</b>	mathematical functions
<b>misc.c</b>	miscellaneous functions
<b>out.c</b>	prints simulation results to output files
<b>resist.c</b>	models photoresist bleaching

---

**1.5 About this Manual**

---

<b>Section 1.0</b>	<b>Introduction</b> A general overview of TEMPEST is provided.
<b>Section 2.0</b>	<b>Installation</b> This section discusses how the simulation program TEMPEST and other auxiliary programs are compiled and installed.
<b>Section 3.0</b>	<b>Tutorial</b> Two tutorial examples to let the user become familiar with the commands and procedures of running TEMPEST are given
<b>Section 4.0</b>	<b>Detail Command Descriptions</b> Detail descriptions of the commands and input file keywords are presented in this section.
<b>Section 5.0</b>	<b>Useful Hints</b> Some useful hints for running TEMPEST are given.
<b>Appendix A</b>	<b>Summary of Commands</b> Important commands are briefly listed.

The conventions used throughout this users' guide are as follows:

- Words which are in ALL CAPITAL LETTERS represent programs or shell scripts.
- Words in **bold** letters represent commands to be typed exactly as it appears on the users' guide.
- Words in *italics* represent commands or arguments which can take on different values or character strings.
- Words inside [square brackets] are options to commands which may be omitted.
- Words in the `Courier` font represent characters in a file or output from programs.
- The string "ws%" represents the unix prompt on a workstation.

## 2.0 Installation

---

### 2.1 Resource Requirements

---

The computing environment of the user should:

- run unix C-shell
- have a C compiler
- have a postscript printer
- have at least 20 mega bytes of disk space available to hold the data files

In addition, it is useful to have X11 windows to display plots of the data by the plotting program PLOTMTV.

### 2.2 Installation

---

This software package consists of the main program TEMPEST as well as other supporting routines to be run on the user's workstation. All the programs except for SPLAT are written in the programming language C.

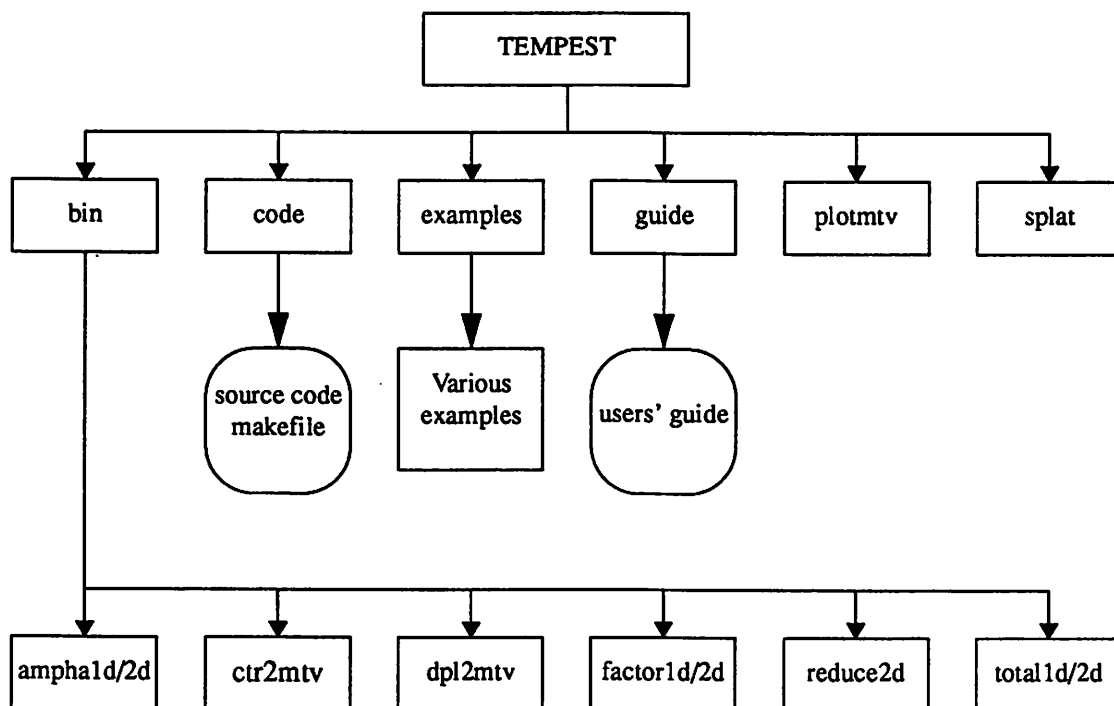
1. TEMPEST (Program for electromagnetic simulation.)
2. AMPHA1D and AMPHA2D (Programs which calculate the amplitudes and phases of the electromagnetic fields from the instantaneous field values.)
3. CTR2MTV and DPL2MTV (Programs which convert CONTOUR and DRAWPLOT data format to PLOTMTV data format.)
4. FACTOR1D and FACTOR2D (Programs which scale the axes of plot files.)
5. REDUCE2D (Program for reducing data file size.)
6. TOTAL1D and TOTAL2D (Programs which calculates the total electric or magnetic field from the values of the x, y, and z components.)
7. PLOTMTV (Program for displaying data files.)
8. SPLAT (Program written in Fortran for calculating aerial images.)

To install TEMPEST, please move to a directory (say *\$HOME/bin*) where TEMPEST should reside. Then insert the tape into the tape drive, and type the following command:

```
ws% tar xv
```

After tape reading is completed, which takes about 5 minutes, a TEMPEST directory is created in the *\$HOME/bin* directory. The programs are organized as follows:





---

## 2.3 Compilation

---

### 2.3.1 TEMPEST

---

To compile TEMPEST, follow the steps:

1. `ws% cd $HOME/bin/TEMPEST/code`
2. `cm% make`

### 2.3.2 SPLAT

---

The program SPLAT can be compiled using make:

1. `ws% cd $HOME/bin/splat`
2. `ws% make`

### 2.3.3 PLOTMTV

---

Compilation of plotmtv uses the utility `xmkmf`. Please following the instructions in the file `$HOME/bin/plotmtv/README.INSTALL`

### **2.3.4 OTHER ROUTINES**

---

The other routines such as AMPHA1D, AMPHA2D, CHECK, FACTOR1D, FACTOR2D, IMAGE2D, and REDUCE2D can be compiled as follows:

1. `ws% cd $HOME/bin/TEMPEST/bin`
2. `ws% makeall`

### **2.4 Testing**

---

When all the programs are compiled, they can be moved to a directory which is contained in the unix variable \$PATH, for example, *\$HOME/bin*.

To test if the programs are installed properly and the computing environment is proper, please do the following:

1. `ws% cd $HOME/bin/tempest/examples`
2. `ws% run.csh`

Several examples are run which illustrate the normal workings of TEMPEST.

3. Follow the examples in Section 3.0 to make sure that all the routines are functioning properly.

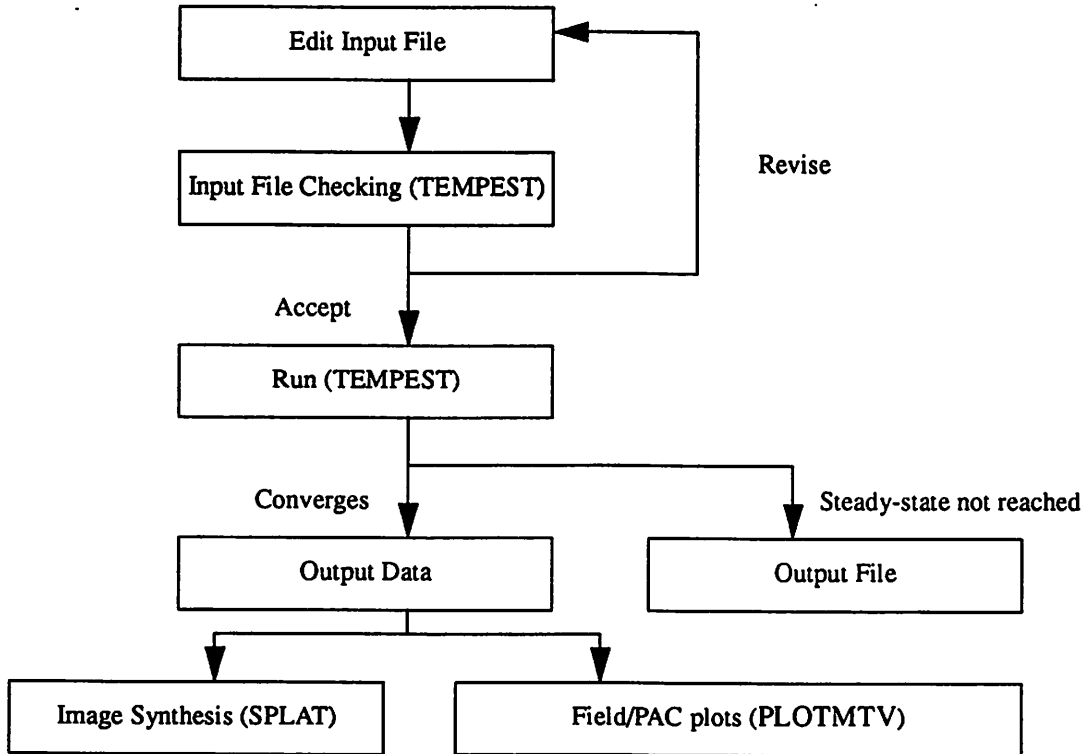
### 3.0 Tutorial

---

#### 3.1 Overview

---

The basic steps in running TEMPEST simulation are shown as follows:



A simulation can take anywhere from a few minutes to several hours depending on the size of the problem. The request of large data files (such as a contour plot of a large area) also requires more simulation time.

#### 3.2 Input File to TEMPEST

---

The input file to TEMPEST contains five types of specifications:

1. simulation domain
2. illumination
3. topography
4. analysis (numerical parameters)
5. data (output specifications)

The input file to TEMPEST can be given any name and the entries in the input file can occur in random order. (The only exception is that the definition of **wavelength** must precede the definition of any **block** property.) The input file to TEMPEST consists of a list of keywords; each keyword may be followed by numbers, words, or other keywords. A detail description of the keywords is found in Section 4.1. Comments can be included in the input file by including the string “/\*” before the comment and the string “\*/” after the comment.

As a first example, transmission through a 1X contact hole is studied. The input file is located in the directory \$HOME/bin/TEMPEST/examples/hole and is named *hole.in*. The dimension of the simulation domain is 1.0  $\mu\text{m}$  by 1.0  $\mu\text{m}$  by 0.25  $\mu\text{m}$ . (The actual simulated volume is 0.5  $\mu\text{m}$  by 0.5  $\mu\text{m}$  by 0.25  $\mu\text{m}$  via the use of symmetry.) The chrome mask has a glass substrate with a layer of 80 nm thick chrome. The square opening on the chrome has a width of 0.25  $\mu\text{m}$  on each side. Illumination is assumed to be normally incident at 248 nm, with the electric field linearly polarized in the x-direction. In this study, the transmitted diffraction harmonics are requested to study the transmission characteristics of the mask. In addition to the diffraction harmonics, the transmitted electric field in the x-direction is also requested.

The input file *hole.inq* is shown as follows:

```
/*
input file name: hole.in
*/

/*
The structure is symmetric with respect to both the x- and y-axes.
*/
sym_model 2

/*
The simulation window is 0.5 by 0.5 by 0.25 micron with 65 simulation nodes
in the x- and y-directions and 32 nodes in the z-direction.
*/
x_node 65
y_node 65
z_node 32
x_dim 0.5
y_dim 0.5
z_dim 0.25
periodic

wavelength 0.248
intensity 1.0

/*
The transmitted diffraction harmonics are calculated and the results are
```

```
stored in the file hole.hrm. hole.spl is a SPLAT input file.
*/
transmitted hole.spl hole.hrm
na 0.5
sigma 0.5
focus 0.0
mag 1

err_tol 0.02

/*
Block 1 is glass with index of (1.5, j0.0).
*/
block 1
figure rectangle
0.0 0.5 0.0 0.5 0.0 0.25
index 1.5 0.0
end_block

/*
Block 2 is air with index of (1.0, j0.0).
*/
block 2
figure rectangle
0.0 0.5 0.0 0.5 0.0 0.10
index 1.0 0.0
end_block

/*
Block 3 is chromium with index of (2.5, j2.0).
*/
block 3
figure rectangle
0.0 0.5 0.0 0.5 0.10 0.18
index 2.5 2.0
end_block

/*
Block 4 represents the contact hole opening (air) with index of 1.0.
*/
block 4
figure rectangle
0.375 0.5 0.375 0.5 0.10 0.18
index 1.0 0.0
end_block

/*
The instantaneous electric field in the x-direction of a plane just
underneath the opening is requested at two instants of the wave cycle.
*/
```

```
plot steady xyplane 0.00 e x node 0 63 0 63 15 hole.xyp.000
plot steady xyplane 0.25 e x node 0 63 0 63 15 hole.xyp.250
```

---

### 3.3 Checking the Input File

---

To check whether the input file is correct, one can execute TEMPEST with the input file name as the only argument:

```
ws% tempest hole.in
```

With this command, the following is displayed, indicating that there is no error in the input file *hole.in*:

```
T E M P E S T
ver4.0
February 7, 1995

Starting TEMPEST...

Reading input file [hole.in]...
Done reading input file [hole.in].

There is no error in the file.
```

---

### 3.4 Running Simulations

---

With no error messages, the input file is in the correct format and the electromagnetic field solver TEMPEST can be run. To execute the program TEMPEST, the user must supply two arguments: the first being the name of the input file, and the second is the name of the output file to which information concerning the simulation is to be written.

The amount of memory needed to run the program is proportional to the number of simulation nodes in the domain, i.e., the product of the number of nodes in the x-, y-, and z-directions. As a rule of thumb, a domain with 128k nodes (64 by 64 by 32) requires about 16MBytes of memory.

To run TEMPEST, type

```
ws% tempest hole.in hole.out
```

```
T E M P E S T
ver4.0
February 7, 1995

Starting TEMPEST...

Reading input file [hole.in]...
Done reading input file [hole.in].
```

Initializing parameters and variables...

Warning:

Minimum number of simulation nodes per wavelength for the topography is 12.70, fewer than 15. Simulation result may be inaccurate.

Done initializing parameters and variables.

Start iteration...

Period #4:

1st try: 100.00% converged.

Period #5:

1st try: 100.00% converged.

2nd try: 100.00% converged.

Period #6:

1st try: 100.00% converged.

2nd try: 100.00% converged.

3rd try: 100.00% converged.

Converges after 6 periods.

Done iteration.

Writing plot file [hole.xyp.000]...

Done writing.

Writing plot file [hole.xyp.250]...

Done writing.

Calculating diffraction efficiencies...

Done calculating diffraction efficiencies.

Writing to output file [hole.out]...

Done writing.

Program executed successfully.

Five files are generated for this simulation:

1. *hole.out* (output file which contains information on the simulation run)
2. *hole.hrm* (harmonic file which contains information on the diffraction harmonics)
3. *hole.spl* (SPLAT input file used for generating the image)
4. *hole.xyp.000* and *hole.xyp.250* (electric field plots across an xy-plane in PLOTMTV format)

### 3.5 Output Data

---

#### 3.5.1 Output File

---

The output file of TEMPEST contains information on the simulation parameters, such as the spatial and temporal discretization values, and recapitulates some of the input parameters. If steady-state is reached and the user requested so, information on the diffraction harmonics is written to the output file and harmonic file as well. This information can be used to synthesize the optical image profile of the structure using the simulation program SPLAT.

The output file generated, *hole.out*, is shown below. It summarizes the topography information and numerical parameters and contains information on the diffraction harmonics.

T E M P E S T

ver4.0

February 7, 1995

University of California, Berkeley

Miscellaneous Information:

Input file name: hole.in  
Output file name: hole.out  
SPLAT input file name: hole.spl  
Harmonic file name: hole.hrm  
There are 2 plot files:  
1. Steady-state field hole.xyp.000  
2. Steady-state field hole.xyp.250

Process Parameters:

Free space wavelength: +2.4800e-01 um  
Incident theta: +0.0000e+00  
Incident phi: +0.0000e+00  
Refl abs angle: +0.0000e+00 +9.5166e+00  
Tran abs angle: +0.0000e+00 +1.4359e+01  
Incident relative (Ex Ey): +1.0000e+00 +0.0000e+00  
Incident cosine (Ex Ey Ez): +1.0000e+00 +0.0000e+00 +0.0000e+00  
Incident cosine (Hx Hy Hz): +0.0000e+00 -1.0000e+00 -0.0000e+00  
Incident cosine (Kx Ky Kz): +0.0000e+00 +0.0000e+00 -1.0000e+00  
Illumination intensity: +1.0000e+00 mW/cm2  
Mask file name: None



Topography:

The structure is assumed to be periodic.

Symmetry: wrt x- and y-axes

x dimension: 5.0000e-01 um

y dimension: 5.0000e-01 um

z dimension: 2.5000e-01 um

Index at north boundary: 1.5000e+00 0.0000e+00

Index at south boundary: 1.0000e+00 0.0000e+00

Number of building blocks: 4

Block 1:

index (+1.5000e+00 +0.0000e+00)

eps (+2.2500e+00 +0.0000e+00)

eta (+2.5115e+02 +0.0000e+00)

cond (+0.0000e+00)

fig rectangle

xl xh 0.0000e+00 5.0000e-01

yl yh 0.0000e+00 5.0000e-01

zl zh 0.0000e+00 2.5000e-01

Block 2:

index (+1.0000e+00 +0.0000e+00)

eps (+1.0000e+00 +0.0000e+00)

eta (+3.7673e+02 +0.0000e+00)

cond (+0.0000e+00)

fig rectangle

xl xh 0.0000e+00 5.0000e-01

yl yh 0.0000e+00 5.0000e-01

zl zh 0.0000e+00 1.0000e-01

Block 3:

index (+2.5000e+00 +2.0000e+00)

eps (+2.2500e+00 +1.0000e+01)

eta (+9.1885e+01 +7.3508e+01)

cond (+6.7251e+05)

fig rectangle

xl xh 0.0000e+00 5.0000e-01

yl yh 0.0000e+00 5.0000e-01

zl zh 1.0000e-01 1.8000e-01

Block 4:

index (+1.0000e+00 +0.0000e+00)

eps (+1.0000e+00 +0.0000e+00)

eta (+3.7673e+02 +0.0000e+00)

cond (+0.0000e+00)

fig rectangle

xl xh 3.7500e-01 5.0000e-01

yl yh 3.7500e-01 5.0000e-01  
zl zh 1.0000e-01 1.8000e-01

Computational parameters:

Distance between nodes (dx): 7.8125e-03 um  
Simulation grid: 65 x 65 x 32  
Minimum nodes per wavelength: 12.70  
Time increment (dt): 1.4772e-17 second  
Iterations per period (wave cycle): 56  
Bulk model: 0  
BC model: 4

Convergence information:

Wave cycles before convergence check: 3  
Maximum iteration wave cycles: 60  
Error tolerance: 2.00%  
Number of periods for convergence: 6

Transmitted diffraction efficiencies:

Number of diffracted orders: 9 9

Real Imag  
...  
order[ +0 ][ -3 ]: +2.490130e-02 +1.142663e-02  
+0.000000e+00 +0.000000e+00  
+0.000000e+00 +0.000000e+00  
-1.546077e-02 -7.116144e-03  
order[ +0 ][ -2 ]: -3.704284e-02 +5.493885e-03  
+0.000000e+00 +0.000000e+00  
+0.000000e+00 +0.000000e+00  
+3.124916e-02 -4.320902e-03  
order[ +0 ][ -1 ]: +4.080810e-02 -1.982616e-02  
+0.000000e+00 +0.000000e+00  
+0.000000e+00 +0.000000e+00  
-3.827879e-02 +1.834613e-02  
order[ +0 ][ +0 ]: -3.222521e-02 +2.422668e-02  
+0.000000e+00 +0.000000e+00  
+0.000000e+00 +0.000000e+00  
+3.181921e-02 -2.402239e-02  
order[ +0 ][ +1 ]: +3.978613e-02 -2.180464e-02  
+0.000000e+00 +0.000000e+00  
+0.000000e+00 +0.000000e+00  
-3.733248e-02 +2.020228e-02  
order[ +0 ][ +2 ]: -3.632597e-02 +9.098263e-03

```
+0.000000e+00 +0.000000e+00
+0.000000e+00 +0.000000e+00
+3.067517e-02 -7.363049e-03
order[ +0 ][ +3 ]: +2.630842e-02 +7.649179e-03
+0.000000e+00 +0.000000e+00
+0.000000e+00 +0.000000e+00
-1.633759e-02 -4.770555e-03
...
Transmitted energy: 2.95%
Total refl and tran energy: 19.72%
```

---

### 3.5.2 Image Synthesis

---

Based on the information of the diffraction harmonics contained in the output file, the optical image profile of the simulated structure can be synthesized given the characteristics of the optical system, including the numerical aperture (NA), partial coherence factor, and the magnification factor. Since TEMPEST assumes coherent illumination, partial coherence is modeled by Hopkins' approach<sup>1</sup>. Two assumptions are made:

1. The magnitudes of the diffraction efficiencies are independent of the angle of illumination.
2. The illumination energy is distributed evenly over the illumination cone.

With these assumptions, the intensity at every point in space can be calculated by evaluating the transmission cross coefficients<sup>1</sup>. This approach can be used to model bright-field or dark-field optical systems, arbitrary lens aberrations as well as defocus effects.

For the example structure, assume that the optical system is 1X and has an NA of 0.5 and a partial coherence factor of 0.5. The optical signal produced by the chrome mask can be synthesized by the command

```
ws% splat < hole.spl
```

where *hole.spl* is generated by TEMPEST. The command *splat* is described in detail in Section 4.3.1.

With the above command, two data files are generated: *hole.spl.i1d* in the DRAWPLOT format and *hole.spl.i2d* in the CONTOUR format. To convert these files into the PLOTMTV format, use DPL2MTV and CTR2MTV:

```
ws% dpl2mtv hole.spl.i1d hole.mtv1d
ws% ctr2mtv hole.spl.i2d hole.mtv2d
```

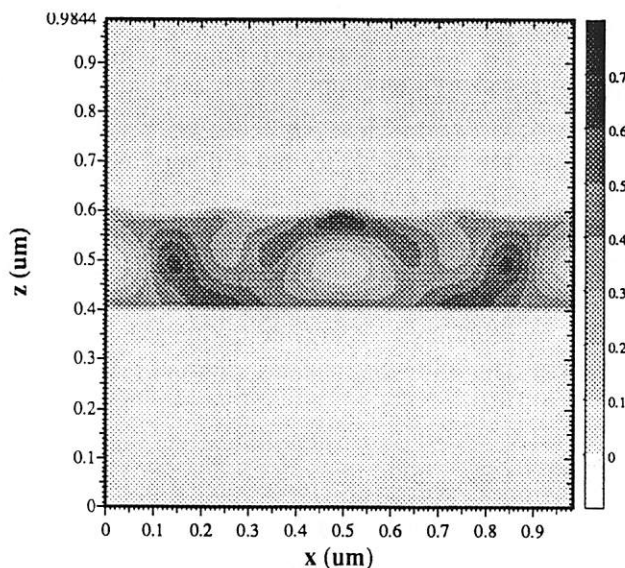
Since the intensity data calculated by SPLAT have negative values when the transmitted image is requested, the PLOTMTV data files must be modified by

multiplying the intensity value by -1. This is accomplished by the routines FACTOR1D and FACTOR2D:

```
ws% factor1d hole.mtv1d hole.fac1d y -1  
ws% factor2d hole.mtv2d hole.fac2d -1
```

The files can then be plotted with PLOTMTV. For example, the image in Figure 1 can be generated with:

```
ws% plotmtv hole.fac2d
```



**Figure 1**

The optical signal created by a 1  $\mu\text{m}$  by 1  $\mu\text{m}$  square opening on a 1X chromium mask. The optical system has an NA of 0.5 and a partial coherence factor of 0.5.

### 3.5.3 Field and PAC plots

The data files generated by TEMPEST are all in the PLOTMTV format. To plot the data files, simply issue the command

```
ws% plotmtv file_name
```

For example, to display the data file *hole.xyp.000*, type

```
ws% plotmtv hole.xyp.000
```

and the picture in Figure 2 is displayed on the screen.

## 4.0 Detail Command Descriptions

---

### 4.1 Input File Keywords

---

The input file to TEMPEST contains five types of specifications:

1. simulation domain
2. illumination
3. topography
4. analysis (numerical parameters)
5. data (output specifications)

The input file to TEMPEST can be given any name and the entries in the input file can occur in random order. (The only exception is the definition of **wavelength** must precede the definition of any **block** property.) Comments can be included in the input file by including the string `/*` before the comment and the string `*/` after the comment. Further, this version (as does the previous version) of TEMPEST performs error checking of the input file and will be able to detect most input errors. The input file to TEMPEST consists of a list of keywords. Each keyword may be followed by numbers, words, or other keywords. The following is a description of the keywords:

#### 4.1.1 Domain

---

##### ***x\_node*** *x\_simulation\_nodes*

*x\_simulation\_nodes* is a number specifying the number of simulation nodes in the x-dimension of the simulation domain. If **no\_de** is specified, then there is no restriction on the value of *x\_simulation\_nodes*. Otherwise, *x\_simulation\_nodes* must be a positive number which is a power of 2 if **sym\_model** is 0. If **sym\_model** is 1 or 2, *x\_simulation\_nodes* must be a positive number which is a power of 2 plus 1 ( $2^n+1$ ). Typically, at least 15 simulation nodes per wavelength are necessary in the optically densest material to give accurate simulation results.

##### ***y\_node*** *y\_simulation\_nodes*

*y\_simulation\_nodes* is a number specifying the number of simulation nodes in the y-direction of the simulation domain. If **no\_de** is specified, then there is no restriction on the value of *y\_simulation\_nodes*. Otherwise, *y\_simulation\_nodes* must be a positive number which is a power of 2 if **sym\_model** is 0 or 1. If **sym\_model** is 2, *y\_simulation\_nodes* must be a positive number which is a power of 2 plus 1 ( $2^n+1$ ).

##### ***z\_node*** *z\_simulation\_nodes*

*z\_simulation\_nodes* is a number specifying the number of simulation nodes in the z-direction of the simulation domain. There is no restriction on the

value of *z\_simulation\_nodes*. In terms of memory usage, a 64 by 64 by 32 nodes simulation domain needs about 16 MBytes of memory. The maximum number of simulation nodes is limited by the available memory of the computing environment.

**x\_dim x\_length**

*x\_length* is the x-dimension of the simulation domain in micro-meter.

**y\_dim y\_length**

*y\_length* is the y-dimension of the simulation domain in micro-meter.

**z\_dim z\_length**

*z\_length* is the z-dimension of the simulation domain in micro-meter. A constraint is placed on the maximum values of *x\_length*, *y\_length*, and *z\_length* in order to maintain a sufficiently high grid density for simulation accuracy and algorithm stability. Typically, at least 15 simulation nodes per wavelength are necessary in the optically densest material to give accurate simulation results. Therefore, the maximum values of *x\_length*, *y\_length*, and *z\_length* are limited by the number of nodes which is in turn limited by the physical resources of the user's computing environment.

**periodic**

keyword **periodic** specifies that the structure under study is a periodic structure. This is the default.

**isolated**

keyword **isolated** specifies that the structure under study is an isolated structure. The default structure type is **periodic**. Presently, isolated structures are not supported.

**1symmetry**

if the structure possesses symmetry in one of the dimensions, specify **1symmetry**. The structure is then assumed to be symmetric in the x-direction (with respect to the y-axis). When **1symmetry** is specified, the argument of *rel\_ex0* must be 1.0 (the default) or 0.0.

**2symmetry**

For a structure which is symmetric with respect to both the x- and y-axes, specify **2symmetry**. Similar to the case in **1symmetry**, the argument of *rel\_ex0* must be 1.0 (the default) or 0.0.

**sym\_model model\_number**

*model\_number* is 0 if there is no symmetry in the structure. *model\_number* is 1 if the structure is symmetric with respect to the x-axis (same as **1symmetry**). For a structure which is symmetric with respect to both the x- and y-axes, *model\_number* is 2 (same as **2symmetry**). The default value of *model\_number* is 0.

---

#### 4.1.2 Illumination

---

**wavelength *lambda***

*lambda* is the free space wavelength of the incident illumination in micrometer. TEMPEST assumes monochromatic illumination. **wavelength** must be defined before definition of any **block** property.

**intensity *value***

*value* is the intensity of the incident illumination in milli-Watt per centimeter squared. The incident field amplitude is determined from *value*, which is given by the expression  $E = \sqrt{2 \times \text{value} \times 10 \times s(x)} \times \eta$  where  $s(x)$  represents intensity modulation specified by the keyword **mask**. If there is no intensity modulation,  $s(x) = 1$ . The factor 10 is a unit-conversion factor.

**inc\_the *theta\_inc*****inc\_phi *phi\_inc***

The illumination direction is specified by the angles *theta\_inc* and *phi\_inc*. The direction of propagation of the incident radiation makes an angle of *theta\_inc* (in degrees) with the z-axis and an angle of *phi\_inc* (in degrees) with the x-axis. The default value for *theta\_inc* and *phi\_inc* is zero. In the current version, oblique incidence works only for periodic structures. Further, only discrete values of *theta\_inc* and *phi\_inc* are allowed, corresponding to integral multiples of 360° phase difference between the east-west and north-south boundaries.

**rel\_ex0 *ex0***

The relative amount of x-polarized electric field is *ex0*, which is a value between 0 and 1. From *ex0*, the relative amount of y-polarized electric field is calculated by  $ey0 = \sqrt{1 - ex0^2}$ . Together with *theta\_inc* ( $\theta$ ) and *phi\_inc* ( $\phi$ ), the directional cosines of  $\vec{E}$ ,  $\vec{H}$ , and  $\kappa$  are calculated to be

$(ex0 \cos \phi + ey0 \sin \phi, -ex0 \sin \phi + ey0 \cos \phi, ey0 \sin \theta)$ ,  
 $(-ex0 \cos \theta \sin \phi + ey0 \cos \phi, -ex0 \cos \theta \cos \phi - ey0 \sin \phi, -ex0 \sin \theta)$ , and  
 $(\sin \theta \sin \phi, \sin \theta \cos \phi, -\cos \theta)$  respectively.

**inc\_abs1 *inc1*****inc\_abs2 *inc2***

The second order absorbing boundary condition implemented allows the user to choose two angles at which the boundary condition absorbs perfectly. On the top boundary, these angles are specified by *inc1* and *inc2* (in degrees). They are measured from the z-axis.

**tra\_abs1 *tra1*****tra\_abs2 *tra2***

Similar to *inc1* and *inc2*, *tra1* and *tra2* are the perfect absorption angles at the bottom boundary in degrees measured from the z-axis.

**mask *mask\_file***

keyword **mask** specifies that the incident intensity is not uniform across the simulation domain. The intensity modulation information is contained in

*mask\_file*, a file in the PLOTMTV format which may be generated from SPLAT and CTR2MTV.

---

### 4.1.3 Topography

---

**block** *block\_num* **figure** *block\_figure* *block\_info* **end\_block**

keyword **block** is one of two ways to specify properties of the simulation structure. **block** specifies that the properties of block *block\_num* are contained in *block\_info*. TEMPEST continues parsing properties for block *block\_num* until the keyword **end\_block** is encountered. The illumination wavelength must be defined before any block information can be given. If two blocks overlap in volume, then the material properties of the overlapped volume is determined by the latter defined block. The maximum number of blocks allowed is 1024.

**figure** signifies that the next word defines the shape of the current block.

*block\_figure* can be one of the following:

**rectangle** *xl xh yl yh zl zh*

**rectangle** specifies that the current block is a rectangular box bounded in the x-direction by [*xl*, *xh*], in the y-direction by [*yl*, *yh*], and in the z-direction by [*zl*, *zh*].

**sphere** *radius xcenter ycenter zcenter*

**sphere** specifies that the current block is a sphere with radius *radius* centered at (*xcenter*, *ycenter*, *zcenter*).

**xcylinder** *radius xl xh ycenter zcenter*

**xcylinder** specifies that the current block is a circular cylinder oriented along the x-direction from *xl* to *xh* with radius *radius*. The circular cross-section on the yz-plane is centered at (*ycenter*, *zcenter*).

**ycylinder** *radius yl yh zcenter xcenter*

**ycylinder** specifies that the current block is a circular cylinder oriented along the y-direction from *yl* to *yh* with radius *radius*. The circular cross-section on the zx-plane is centered at (*zcenter*, *xcenter*).

**zcylinder** *radius zl zh xcenter ycenter*

**zcylinder** specifies that the current block is a circular cylinder oriented along the z-direction from *zl* to *zh* with radius *radius*. The circular cross-section on the xy-plane is centered at (*xcenter*, *ycenter*).

**xwedge** *xl xh yl zl y2 z2 y3 z3*

**xwedge** specifies that the current block is a wedge with a triangular interface with the yz-plane defined by the coordinates (*yl*, *zl*), (*y2*, *z2*), and (*y3*, *z3*). The wedge is oriented along the x-direction from *xl* to *xh*.



**ywedge** *yl yh zl xl z2 x2 z3 x3*

**ywedge** specifies that the current block is a wedge with a triangular interface with the *zx*-plane defined by the coordinates (*zl*, *xl*), (*z2*, *x2*), and (*z3*, *x3*). The wedge is oriented along the *y*-direction from *yl* to *yh*.

**zwedge** *zl zh xl yl x2 y2 x3 y3*

**zwedge** specifies that the current block is a wedge with a triangular interface with the *xy*-plane defined by the coordinates (*xl*, *yl*), (*x2*, *y2*), and (*x3*, *y3*). The wedge is oriented along the *y*-direction from *zl* to *zh*.

**block\_info** consists of a list of numbers and words, which may be:

**index** *real\_index imag\_index*

*real\_index* and *imag\_index* are respectively the real and imaginary parts of the refractive index of the current block of material. *imag\_index* must be either greater than zero (representing dielectric loss) or equal to zero (lossless dielectric). If the value of *imag\_index* is greater than that of *real\_index*, “dispersive” must be specified in the input file.

**eps** *real\_eps imag\_eps*

*real\_eps* and *imag\_eps* are respectively the real and imaginary parts of the relative permittivity of the current block of material. *imag\_eps* must be greater than zero (representing dielectric loss) or equal to zero (lossless dielectric). For each block, either **index** or **eps**, but not both, should be specified.

**cond** *conductivity*

if the current block is a metal, the properties can be specified by its conductivity *conductivity*. If only **cond** is specified, the real part of the relative permittivity is assumed to be 1.0 and the imaginary part is calculated from the value of *conductivity*. If both **cond** and **eps** are specified, the real part of the dielectric constant is given by *real\_eps*, and the imaginary part is given by the sum of contributions from *imag\_eps* and *conductivity*. For any block, **index** and **cond** cannot both be specified.

**resist** *conc a b c dose step*

**resist** specifies that the current block has dynamic change with respect to exposure. *conc* is the initial PAC value of the photoresist block. *a* is Dill’s bleachable absorption coefficient of the photoresist in per micro-meter. *b* is Dill’s unbleachable absorption coefficient of the photoresist in per micro-meter. *c* is the bleach rate of the photoresist model proposed by Dill in centi-meter squared per milli-Joule. *dose* is the total exposure energy of the photoresist in milli-Joule per centi-meter squared. The total *dose* is divided into *step*

number of smaller doses, the sum of which equals the total *dose*. For each dose step, the steady-state electric field is computed and the bleaching of the photoresist block is modeled using Dill's ABC parameters.

**read\_index** *index\_file*

Another way to specify the properties of the simulation structure is by the keyword **read\_index**. This loads the real and imaginary parts of the refractive index at every simulation node position from the file *index\_file*. The *index\_file* is consisted entirely of pairs of numbers. The number of pairs must be identical to the product of *x\_simulation\_nodes*, *y\_simulation\_nodes* and *z\_simulation\_nodes*. The first number of each pair is the real part of the refractive index and the second number is the imaginary part. The order of the pairs is such that the first pair corresponds to the lower left corner of the simulation domain. Ordering is done by first incrementing along the x-direction, followed by the y-direction, and lastly the z-direction. A file of such format can be generated using the keyword **write\_index**.

**write\_index** *index\_file*

keyword **write\_index** specifies that the real and imaginary parts of the refractive index at every simulation node position to be written to the file *index\_file*. This file can be used as the argument of **read\_index** for other TEMPEST simulations.

**dispersive**

if the structure contains highly dispersive materials, i.e.,  $n_i > n_r$ , the conventional algorithm is unstable. A special routine must be used by specifying **dispersive** in the input file.

---

**4.1.4 Analysis**

---

**reflected**

keyword **reflected** specifies that the reflected diffraction harmonics are calculated. The diffraction harmonics are computed at discrete scattering angles which depend on the incident wavelength and the dimensions of the simulation domain. These diffraction harmonics can be used by the program SPLAT to generate an optical image profile of the simulated structure.

**transmitted**

keyword **transmitted** specifies that the transmitted diffraction harmonics are calculated.

**no\_de**

keyword **no\_de** specifies that no diffraction harmonics are calculated. This is the default.

**min\_cycle** *min\_wave\_cycles*

Since checking for steady-state slows down the simulation, TEMPEST excites the simulation domain *min\_wave\_cycles* wave cycles before checking

for steady-state. Typically, at least 20 wave cycles of excitation are required before steady-state is reached. The number of wave cycles needed increases with complexity of the topography as well as the *z\_length* of the simulation domain. If unspecified, the program calculates *min\_wave\_cycles* automatically.

**max\_cycle** *max\_wave\_cycles*

Simulation continues until steady-state is reached or until the domain has been excited with *max\_wave\_cycles* wave cycles, whichever comes first. If steady-state is not reached before *max\_wave\_cycles* wave cycles of excitation, diffraction efficiencies are not calculated. A typical value for *max\_wave\_cycles* is 100. If unspecified, the program calculates *max\_wave\_cycles* automatically.

**err\_tol** *fraction*

Steady-state is determined by comparing the field values across an xy-plane (set at 5 simulation nodes from the boundary where the diffraction harmonics are taken, if *no\_de* is specified, the plane is set at 5 simulation nodes from the top boundary) at the same instant in successive wave cycles. If the variation of the electric field at each node on the plane is less than *fraction* times a comparing value (this comparing value is determined by the specification or no specification of the keyword *err\_rel*) for three consecutive wave cycles, steady-state is reached. A typical value for *fraction* is 0.02, where:

$$\text{error} = \text{fraction} \times \text{amplitude} \\ |E_{\text{new}} - E_{\text{old}}| \geq \text{error}$$

**err\_rel**

*err\_rel* specifies that the comparing value is the instantaneous field value. If *err\_rel* is not specified, the incident field amplitude is used as the comparing value.

---

#### 4.1.5 Data

---

**plot** *plot\_var plot\_type [fraction] [plot\_dir] coor [min1 max1 min2 max2 min3 max3] plot\_file*

*plot\_var* must be one of the five keywords: **pac**, **refractive**, **steady**, **nonsteady**, or **transient**.

**pac** requests a PAC concentration plot after the photoresist is exposed.

**refractive** requests a plot of the real part of the refractive index. This is a check to ensure that the structure is loaded into TEMPEST correctly.

**steady** requests a plot of the instantaneous values of a field component at steady-state. The field component is specified by *plot\_dir* and the instant of the wave cycle is specified by *fraction*.

**nonsteady** requests a plot of the instantaneous values of a field component before the field reaches steady-state. The field component is specified by *plot\_dir*, and *fraction* specifies the number of cycles after the initial excitation at which the field values are taken. Note that *instant* in this case must be smaller than *min\_wave\_cycles* specified in *min\_cycle*.

**transient** requests a transient plot of a field component specified by *plot\_dir*. This option traces the time development of the field value of a point in the simulation domain. When **transient** is specified, *plot\_type* must be **point**.

*plot\_type* must be one of the six keywords: **point**, **xline**, **yline**, **zline**, **xyplane**, **yzplane**, or **zxplane**.

**point** requests a plot of *plot\_var* at a point in the simulation domain. This *plot\_type* is usually used when a **transient** plot is requested.

**xline** requests a line plot of *plot\_var* at a cutline in the x-direction from *min1* to *max1* at the position  $y=\text{min2}$  and  $z=\text{min3}$ . *max2* and *max3* are not specified in this case.

**yline** requests a line plot of *plot\_var* at a cutline in the y-direction from *min1* to *max1* at the position  $z=\text{min2}$  and  $x=\text{min3}$ . *max2* and *max3* are not specified in this case.

**zline** requests a line plot of *plot\_var* at a cutline in the z-direction from *min1* to *max1* at the position  $x=\text{min2}$  and  $y=\text{min3}$ . *max2* and *max3* are not specified in this case.

**xyplane** requests a two-dimensional plot of *plot\_var* across an xy-plane bounded by  $x=\{\text{min1},\text{max1}\}$  and  $y=\{\text{min2},\text{max2}\}$  at the position  $z=\text{min3}$ . *max3* is not specified in this case.

**yzplane** requests a two-dimensional plot of *plot\_var* across an yz-plane bounded by  $y=\{\text{min1},\text{max1}\}$  and  $z=\{\text{min2},\text{max2}\}$  at the position  $x=\text{min3}$ . *max3* is not specified in this case.

**zxplane** requests a two-dimensional plot of *plot\_var* across an zx-plane bounded by  $z=\{\text{min1},\text{max1}\}$  and  $x=\{\text{min2},\text{max2}\}$  at the position  $y=\text{min3}$ . *max3* is not specified in this case.

*fraction* is specified only if a **steady** or **nonsteady** plot is desired. If a **steady** plot is desired, *fraction* is a number between 0.0 and 1.0 specifying the fraction of the wave cycle at which the plot is generated. A value of 0.0 means the beginning of a wave cycle and a value of 0.25 means a quarter of a wave cycle after the beginning (a phase of 90 degrees). If a **nonsteady** plot is desired, a plot after

*fraction* cycles of excitation is generated. Note that for a **nonsteady** plot, the value of *fraction* must be a number between 0.0 and *min\_wave\_cycles*.

*plot\_dir* is specified only if *plot\_var* is one of **steady**, **nonsteady**, or **transient**. *plot\_dir* contains two entries. The first entry is either **e** or **h**, specifying the electric field or the magnetic field. The second entry can be one of the following: **x**, **y**, or **z**, specifying the direction of the field.

*coor* can be either of the two keywords **position** or **node**. **position** specifies that the coordinates *min1*, *max1*, *min2*, *max2*, *min3*, and *max3* are specified in micro-meters. **node** specifies that *min1*, *max1*, *min2*, *max2*, *min3*, and *max3* correspond to the simulation node number. The convention is that (0, 0, 0) corresponds to the spatial location (0, 0, 0) and (*x\_simulation\_nodes*, *y\_simulation\_nodes*, *z\_simulation\_nodes*) corresponds to (*x\_length*, *y\_length*, *z\_length*).

*plot\_file* is the name of the file to which data are written. All the data files are in the PLOTMTV format. A maximum of 50 plots can be generated.

---

## 4.2 TEMPEST commands

---

### 4.2.1 TEMPEST

---

The program TEMPEST takes two arguments. The first of which is the name of the input file which contains the topography information. The second argument is the name of the output file to which information on the simulation run is written.

```
cm% tempest input_file output_file
```

---

## 4.3 Post-TEMPEST Workstation Commands

---

### 4.3.1 SPLAT

---

Based on the information of the diffraction harmonics contained in the output file, the optical image profile of the simulated structure can be synthesized given the characteristics of the optical system including the numerical aperture (NA), partial coherence factor ( $\sigma$ ), and the magnification factor. This is done by the simulation program SPLAT which calculates the aerial images. Since TEMPEST assumes coherent illumination, partial coherence in SPLAT is modeled by Hopkins' approach<sup>1</sup>. Two assumptions are made:

1. The magnitudes of the diffraction efficiencies are independent of the angle of illumination.
2. The illumination energy is distributed evenly over the illumination cone.

With these assumptions, the intensity at every point in space can be calculated by evaluating the transmission cross coefficients<sup>1</sup>. This approach can be used to model bright-field or dark-field optical systems, arbitrary lens aberrations as well as defocus effects.

For a detailed description of SPLAT, please refer to the SPLAT User's Guide<sup>5</sup>. A typical input file for SPLAT is shown as follows:

```
# .
# hole.spl
#
1: printlevel = 3;
2: lambda = 0.248;
3: NA = 0.500;
31: magnification = 1.000 mode = 1;
4: defocus = 0.000;
5: sigma = 0.500;
6: mask = 0.500um x 0.500 um @0;
37: "hole.hrm ";
10:
11: 0.000 0.000 1.000 1.000 51 51 'hole.spl.i2d';
14: 0.000 0.000 1.000 0.000 201 'hole.spl.i1d';
0:
```

---

### 4.3.2 PLOTMTV

---

PLOTMTV is developed by Kenny Toh at Intel Corporation. It is available in the public domain and is distributed along with TEMPEST. Files in the PLOTMTV format can be plotted using the command:

```
ws% plotmtv [-nodate] [-colorps] [-l] file_name
```

where *file\_name* is the name of the data file. If [-nodate] is specified, the postscript file generated by PLOTMTV has not date stamp on it. To generate color postscript files, specify the [-colorps] option. The [-l] option specifies that the graph is plotted in the landscape orientation.

---

### 4.3.3 AMPHA1D and AMPHA2D

---

The amplitudes and phases of the electromagnetic can be calculated from the instantaneous field values using the routines AMPHA1D and AMPHA2D. These two routines take as input two data files corresponding to the instantaneously field values at two instants separated by a quarter of a wave cycle. Two data files are generated which contain respectively the amplitude and phase information.

The synopses of **ampha1d** and **ampha2d** are identical. For one-dimensional data files (i.e., data files generated with *plot\_type* equal to *xline*, *yline*, or *zline*), **ampha1d** is used. For two-dimensional data files (i.e., data files generated with *plot\_type* equal to *xyplane*, *yzplane*, or *zxplane*), **ampha2d** is used.

**ws% ampha1d/2d file\_000 file\_250 file\_amp file pha**

where *file\_000* and *file\_250* are the input data files. *file\_250* contains field values which occur a quarter of a wave cycle later than the data in *file\_000*. *file\_amp* and *file pha* contain the amplitude and phase information respectively. The data files *file\_000* and *file\_250* can be generated using the TEMPEST input keyword **plot** as described in Section 4.1.5. For an example, please refer to the TEMPEST input file shown in Section 3.2.

---

#### 4.3.4 FACTOR1D and FACTOR2D

---

In some cases the user may wish to scale an axis of a plot file in the PLOTMTV format. For example, the user may wish to normalize the optical image profile to a certain level. This is accomplished by the routines **FACTOR1D** and **FACTOR2D**.

For line plots, use

**ws% factor1d input\_file output\_file dir factor**

where *input\_file* is the data file in PLOTMTV format and *output\_file* is the name of the file to which the modified data is written. *dir* is either *x* or *y*, specifying either the x-axis or the y-axis is scaled by a factor of *factor*.

For two-dimensional plots, use

**ws% factor2d input\_file output\_file factor**

where *input\_file* is the data file in PLOTMTV format and *output\_file* contains the modified data. All the data points are multiplied by a factor of *factor*.

---

#### 4.3.5 REDUCE2D

---

Sometimes the size of the data file is so large that processing is extremely slow and takes up too much disk space. One way to speed up processing and reduce storage memory is by reducing the number of data points with the routine **reduce2d** as follows:

**ws% reduce2d input\_file output\_file x\_factor y\_factor**

In the x-direction, **reduce2d** takes one datum out of every *x\_factor* data from *input\_file*. In the y-direction, the program takes one datum out of every *y\_factor* data from *input\_file*. The reduced data set is written to *output\_file*. The size of *output\_file* is thus (*x\_factor*\**y\_factor*) times smaller than *input\_file*. From

experience, taking one datum out of two to four data from a raw TEMPEST data file retains the resolution yet improves processing time and reduces storage memory.

---

#### 4.3.6 TOTAL1D and TOTAL2D

---

In some applications the user may wish to have information on the total electric or magnetic field rather than the values of the field components. To calculate the total field from the field component values, TOTAL1D and TOTAL2D, which are applicable for line plots and contour plots respectively, can be used.

*ws% total1d/2d x\_component y\_component z\_component total*

where *x\_component*, *y\_component*, *z\_component* are PLOTMTV data format files containing the field component values in the x-, y-, and z-directions, and *total* is the output file name.

---

#### 4.3.7 DPL2MTV and CTR2MTV

---

The data files generated by SPLAT are either in the DRAWPLOT (in the case of line plots) or the CONTOUR (in the case of contour plots) format. The routines DPL2MTV and CTR2MTV can be used respectively to convert from the DRAWPLOT format to the PLOTMTV format and from the CONTOUR format to the PLOTMTV format:

*ws% dpl2mtv DRAWPLOT\_file PLOTMTV\_file*  
*ws% ctr2mtv CONTOUR\_file PLOTMTV\_file*



## 5.0 Useful Hints

---

### 5.1 Top and Bottom of the Simulated Structure

---

The materials at the top and bottom of the simulated domain must be homogeneous. Also, the entire simulation domain must be filled (material properties specified). It is suggested that the first block of material is defined to be a rectangular box with dimensions conforming to the size of the simulation domain.

### 5.2 Material Interface from the Top and Bottom Boundaries

---

If a material interface exists very close to the top of the simulation domain, simulation results may be inaccurate. The distance between an interface and the top of the simulation domain should be about one wavelength. This consideration applies to the interface closest to the bottom of the simulation domain as well. Also, material properties must be uniform across the top and bottom boundaries of the simulation domain.

### 5.3 Verification of Loaded Topography

---

If it is suspected that the loaded topography does not coincide with which the user has in mind, the user can verify the loaded topography by plotting the real part of the refractive index using the keyword `plot` as described in Section 4.1.5.

### 5.4 Real and Imaginary Parts of the Refractive Index

---

Numerical instability may occur when the imaginary part of the refractive index of a particular material block is greater than the real part. Such a situation may arise if a metal block is present. In such instances, the line “dispersive” must be included in the input file to TEMPEST.

### 5.5 Number of Simulation Nodes

---

The number of simulation nodes needed in order to give accurate simulation results can be determined by the following relation:

$$x\_simulation\_nodes > (15 * largest\_index * x\_length) / wavelength$$

and the number of simulation nodes in the y- and z-directions can be determined from  $x\_simulation\_nodes$  by the ratios of  $x\_length$ ,  $y\_length$  and  $z\_length$ .

---

## 5.6 Periodic and Isolated Structures

---

In the current version, **isolated** structures cannot be analyzed. **periodic** structures must be assumed. Isolated structures can be modeled by using a large periodic simulation domain.

---

## 5.7 Convergence Problems

---

If the user encounters convergence problems, it may be due to the presence of a highly dispersive material block. Specify the keyword **dispersive** may alleviate the problem.

---

**References**

---

- 1 Born & Wolf, *Principles of Optics*, p. 513, Pergamon Press, 1975.
- 2 T. Doi, K. Tadros, B. Kuyel, and A. R. Neureuther, "Edge-profile, Materials and Protective Coating Effects on Image Quality," *Proc. SPIE, Integrated Circuit Metrology, Inspection and Process Control V*, vol. 1464, Mar. 1991.
- 3 Roberto Guerrieri, Karim H. Tadros, John Gamelin, and Andrew Neureuther, "Massively Parallel Algorithms for Scattering in Optical Lithography," *IEEE Trans. on CAD*, vol. 10, no. 9, Sep. 1991.
- 4 W. G. Oldham, S. N. Nandgaonkar, A. R. Neureuther, and M. M. O'Toole, "A General Simulator for VLSI Lithography and Etching Processes: Part I - Application to Projection Lithography," *IEEE Trans. on Electron Devices*, vol. ED-26, no. 4, pp. 717-722, Apr. 1979.
- 5 *SPLAT v5.0 Users' Guide*, Memorandum no. M95/13, ERL, University of California, Berkeley, 1995.
- 6 K. Tadros, A. R. Neureuther, J. Gamelin, and R. Guerrieri, "Investigation of Reflective Notching with Massively Parallel Simulation," *Proc. SPIE, Optical/Laser Microlithography III*, vol. 1264, pp. 322-332, Mar. 1990.
- 7 K. Tadros, A. R. Neureuther, and R. Guerrieri, "Understanding Metrology of Polysilicon Gates through Reflectance Measurements and Simulation," *Proc. SPIE, Integrated Circuit Metrology, Inspection and Process Control V*, vol. 1464, Mar. 1991.
- 8 *CMS Technical Summary*, Thinking Machines Corporation, Nov. 1992.
- 9 Kenny Toh, "Two-Dimensional Images with Effects of Lens Aberrations in Optical Lithography," *M. S. Thesis*, Memorandum No. UCB/ERL M88/30, University of California, Berkeley, May 1988.
- 10 Kenny Toh, "Algorithms for Three-Dimensional Simulation of Photoresist Development," Memorandum. No. UCB/ERL M90/123, *Ph.D. Dissertation*, University of California, Berkeley, December 14, 1990
- 11 A. Wong, T. Doi, D. Dunn, and A. R. Neureuther, "Experimental and Simulation Studies of Alignment Marks," *Proc. SPIE, Optical/Laser Microlithography IV*, vol. 1463, Mar. 1991.
- 12 Alfred K. Wong, and Andrew R. Neureuther, "Polarization Effects in Mask Transmission", *Proc. SPIE, Optical/Laser Microlithography*, Mar. 1992.
- 13 Alfred K. Wong, and Andrew R. Neureuther, "Edge Effects in Phase-shifting Masks for 0.25  $\mu\text{m}$  Lithography," *Proc. SPIE*, vol. 1809, pp. 222-228, 1992.
- 14 Alfred K. Wong, and Andrew R. Neureuther, "Mask Topography Effects in Projection Printing of Phase-Shifting Masks," *IEEE Trans. Elec. Dev.*, vol. 41, no. 6, pp. 895-902, June 1994.

---

**Appendix A Summary of Commands**

---

**A**

**ampha1d/2d**, routines which calculate the amplitudes and phases from instantaneous field data 33

**C**

**ctr2mtv**, data conversion program 35

**D**

**dpl2mtv**, data conversion program 35

**F**

**factor1d/2d**, scaling programs for files in PLOTMTV format 34

**I**

**image2d**, image synthesis routine 32

**P**

**plotmtv**, plotter 33

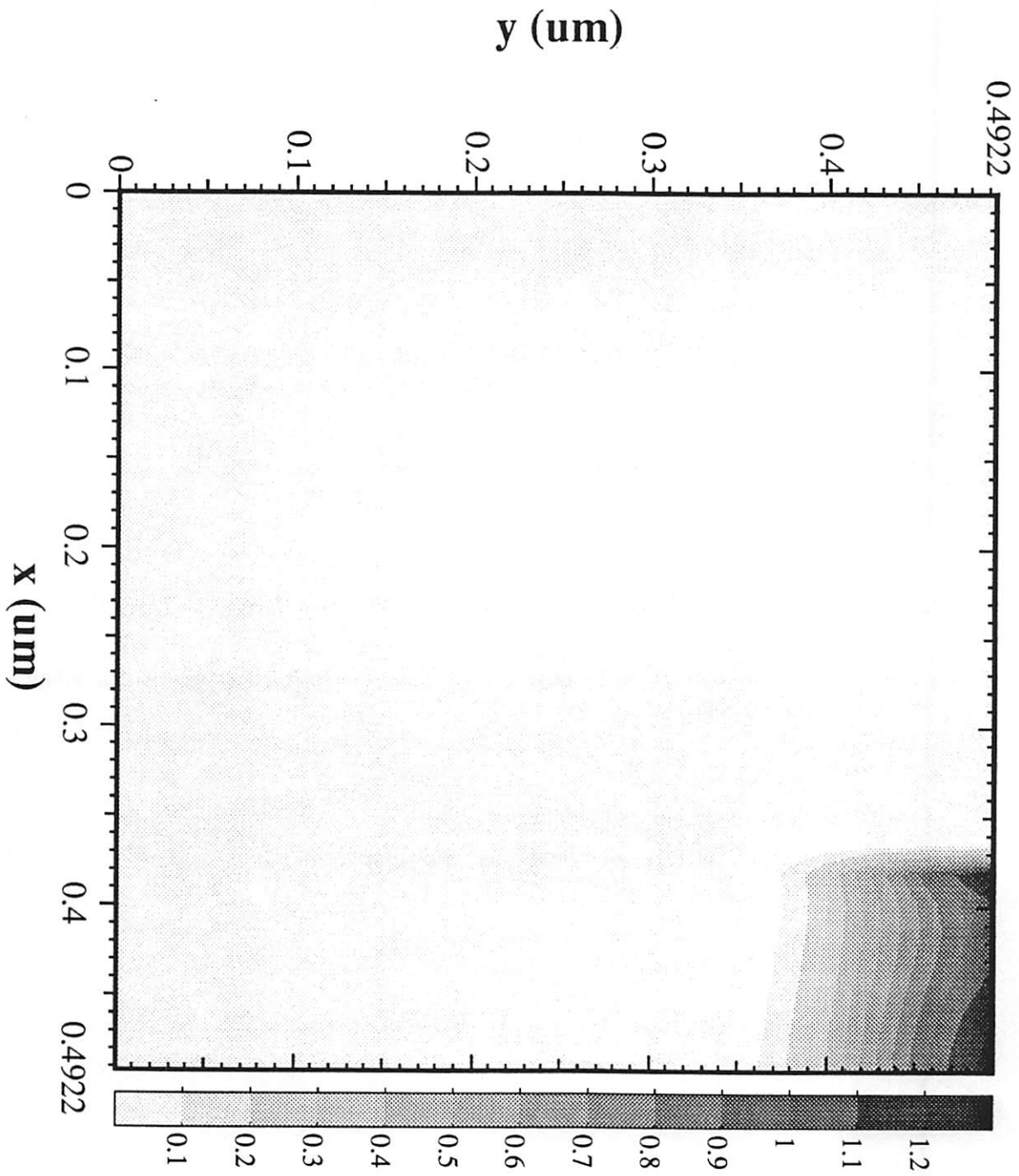
**R**

**reduce2d**, routine which converts large data files to smaller ones by discarding data points 34

**T**

**tempest**, electromagnetic fields solver 32

**total1d/2d**, calculate total field from field components 35



Offending Command = InitDictCount

Error = nametype : undefined

this name is not defined in a dictionary

Stack =

3