

Copyright © 1994, 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**

by

Alfred Wong

Memorandum No. UCB/ERL M94/64

14 July 1994

COVER PAGE

# **TEMPEST USERS' GUIDE**

by

Alfred Wong

Memorandum No. UCB/ERL M94/64

14 July 1994

## **ELECTRONICS RESEARCH LABORATORY**

College of Engineering  
University of California, Berkeley  
94720

# **TEMPEST USERS' GUIDE**

by

Alfred Wong

Memorandum No. UCB/ERL M94/64

14 July 1994

## **ELECTRONICS RESEARCH LABORATORY**

College of Engineering  
University of California, Berkeley  
94720

---

**Table of Contents**

---

Contributors .....	iii
1.0 Introduction.....	1
1.1 Overview.....	1
1.2 New in Version 3.0 .....	2
1.3 Applications Information.....	3
1.4 Source Code Information.....	4
1.5 About this Manual .....	4
2.0 Installation.....	6
2.1 Resource Requirements .....	6
2.2 Installation .....	6
2.3 Compilation .....	7
2.4 Testing.....	8
3.0 Tutorial .....	9
3.1 Overview.....	9
3.2 Input File to TEMPEST.....	9
3.3 Checking the Input File .....	12
3.4 Running Simulations .....	12
3.5 Output Data.....	16
3.6 Example 2: Photoresist Exposure .....	21
4.0 Detail Command Descriptions .....	28
4.1 Input File Keywords .....	28
4.2 TEMPEST commands .....	36
4.3 Pre-TEMPEST Workstation Commands .....	40
4.4 Post-TEMPEST Workstation Commands.....	41

---

## Table of Contents

---

5.0 Useful Hints .....	44
5.1 Top and Bottom of the Simulated Structure .....	44
5.2 Material Interface from the Top and Bottom Boundaries .....	44
5.3 Verification of Loaded Topography .....	44
5.4 Real and Imaginary Parts of the Refractive Index .....	44
5.5 Number of Simulation Nodes .....	44
5.6 Periodic and Isolated Structures .....	45
5.7 Convergence Problems .....	45
References .....	46
Summary of Commands .....	47
Connection Machine Facilities .....	48

---

**Contributors**

---

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

ver1.0 July 23, 1991.

ver2.0 December 31, 1992.

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 and 3.0)

Contributors to earlier releases:

Takeshi Doi(ver1.0)

Eric Hall(Manual Format)

TEMPEST is being 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, Alfred Wong added three-dimensional analysis capabilities on the CM-5 and linked TEMPEST with SPLAT.

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

---

## Contributors

---



# TEMPEST version 3.0

---

**Alfred Wong**

**Electronics Research Laboratory**

**University of California, Berkeley**

---

## **1.0 Introduction**

---

### **1.1 Overview**

---

The acronym TEMPEST<sup>4</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. It 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>. The simulation domain may represent periodic, isolated, 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>5</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 on a workstation using the pre-processor CHECK. This file can then be transferred using telnet or the unix command ftp to one of the computers facilities offering time on a CM. The input geometry is then loaded onto the processors of the CM and 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 locally on a work station by the program IMAGE2D which calls 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 locally using the plotting package PLOTMTV.

The prohibitive expense of high performance computers, such as the CM, causes some concern of availability. However, any user obtaining an account on one of several public access sites around the United States is capable of running simulations from a local work station communicating via the Internet. Appendix B lists several of these sites and the appropriate person to contact for information on how to obtain an account. Note that these sites may have restrictions on the usage of their facilities. Details of the CM can be found in the CM-5 Technical Summary<sup>8</sup>.

---

## **1.2 New in Version 3.0**

---

Version 3.0 of the program TEMPEST has several key extensions from the previous version (version 2.0). Three-dimensional problems can be analyzed in addition to two-dimensional problems. Dispersive materials which cause numerical instability in version 2.0 can be modeled in version 3.0. Image synthesis is performed with a link to the simulation program SPLAT. This link enables the study of rigorous electromagnetic simulation with modified illumination, pupil filtering, and arbitrary lens aberrations simultaneously.

In version 2.0, data are available on the magnetic field as well as the electric field, non-steady-state fields and steady-state fields, transient fields, field amplitudes and instantaneous field values, PAC concentration, and the diffraction harmonics. Besides these data, version 3.0 can provide information on the refractive index, the total field, as well as the value of the Poynting vector.

Display of the data has also been changed in version 3.0. The previous version (version 2.0) uses different plotting programs such as DRAWPLOT and MATHEMATICA for different types of plots. In version 3.0, only one plotting routine (PLOTMTV) is used regardless of the data type. This makes the display of data easier for the user.

There is also a slight change in the way one submits a CM-5 job. After telnet or remote login to the partition manager of the CM-5, the user must issue the command `jrun` instead of `cmattach` to access a partition of the CM-5. To run TEMPEST simulations as batch jobs, one uses the command `jsub` instead of the unix command `at`.

Besides the main electromagnetic fields solver program, TEMPEST version 3.0 includes several auxiliary programs. The program CHECK allows the user to check for possible errors in the input file. The programs AMPHA1D and AMPHA2D allow the user to calculate the amplitude and phase of any electromagnetic field component along any line or plane. IMAGE2D is a routine which calls the simulation program SPLAT for image synthesis. 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>2,4,6,7,11,12,13,14</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 4.4.1. 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, the steady-state field, the total field, the Poynting vector, 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 is written in the C\* programming language. The source code is distributed into nine text files. Four header files are also included. The header files are:

<b>definition.h</b>	constant definitions
<b>global.h</b>	global variable definitions
<b>library.h</b>	libraries to include
<b>patchlevel.h</b>	version and release date information

The files which contain the source code are:

<b>compute.cs</b>	computes the diffraction harmonics
<b>in.cs</b>	reads information from the input file
<b>init.cs</b>	initialization of variables
<b>iterate.cs</b>	fields computation by iteration
<b>main.cs</b>	program control
<b>math.cs</b>	mathematical functions
<b>misc.cs</b>	miscellaneous functions
<b>out.cs</b>	prints simulation results to output files
<b>resist.cs</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.

**Section 5.0****Useful Hints**

Some useful hints for running TEMPEST are given.

**Appendix A****Summary of Commands**

Important commands are briefly listed.

**Appendix B****Connection Machine Facilities**

Use this section as a reference to connection machine facilities.

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.
- The string "cm%" represents the unix prompt on a partition manager of a CM.

---

## 2.0 Installation

---

---

### 2.1 Resource Requirements

---

The local computing environment of the user should:

- run unix C-shell
- have a C compiler
- have access to telnet ftp
- 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

---

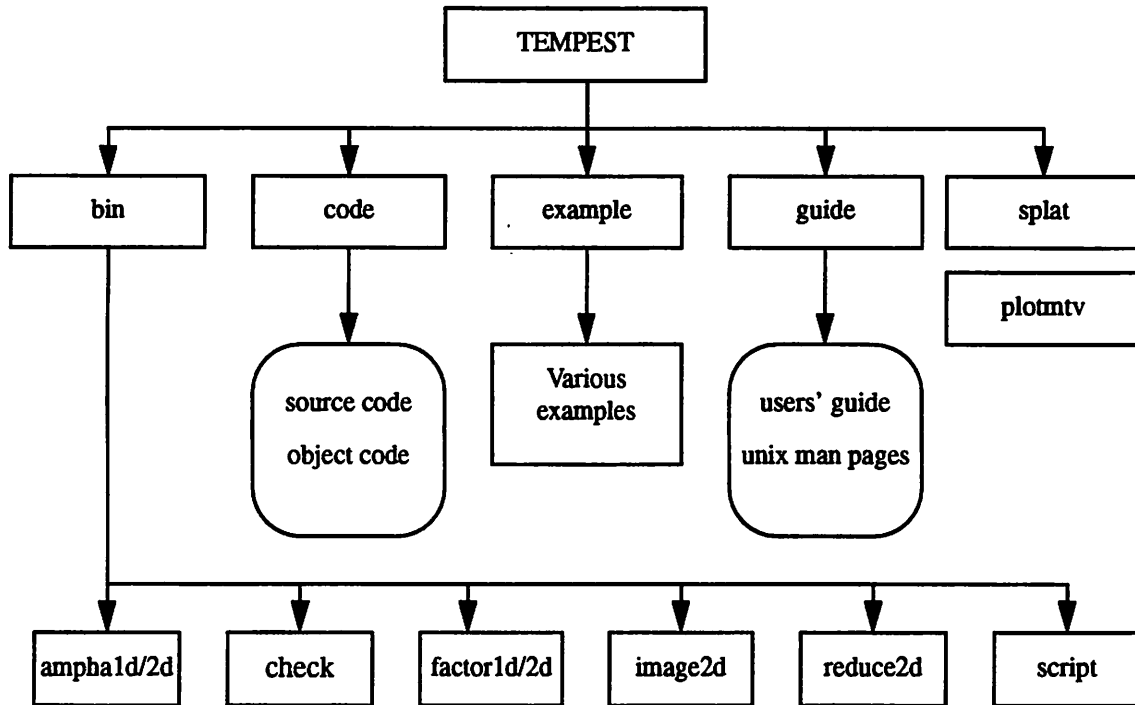
This software package consists of the main program TEMPEST, for use on a CM, as well as other supporting routines to be run on the user's workstation.

1. TEMPEST (CM program in C\* for electromagnetic simulation.)
2. CHECK (Workstation program in C for checking the input file.)
3. IMAGE2D (Workstation program in C which calls SPLAT for image synthesis.)
4. AMPHA1D and AMPHA2D (Workstation programs in C which calculates the amplitudes and phases of the electromagnetic fields from the instantaneous field values.)
5. FACTOR1D and FACTOR2D (Workstation programs in C which scale the axes of plot files.)
6. REDUCE2D (Workstation program in C for reducing data file size.)
7. PLOTMTV (Workstation program written in C for displaying data files.)
8. SPLAT (Workstation program written in C 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 10 minutes, a TEMPEST directory is created in the *\$HOME/bin* directory. The programs and shell scripts are organized as follows:



---

## **2.3 Compilation**

---

### **2.3.1 TEMPEST**

---

The electromagnetic simulation program TEMPEST must be installed and run on a partition of the CM. To compile TEMPEST, follow the steps:

1. **ws% cd *\$HOME/bin/TEMPEST/code***
2. use the unix command **ftp** or **rcp** to copy every file in the directory *\$HOME/bin/TEMPEST/code* to a directory (e.g., *CM\_code*) on the partition manager of the CM.
3. **telnet** or **rlogin** to a partition manager of the CM
4. **cm% cd *CM\_code***
5. **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/example**
2. **ws% test**

The pictures shown in Figure 1 to Figure 4 of this manual should be displayed on the screen within a few seconds. If a plot is not displayed, check that the **\$DISPLAY** variable is set properly (**ws% setenv DISPLAY host:display**) and the machine you are working on is added as an **xhost** to the work station machine (**ws% xhost machine**).

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



---

### **3.0 Tutorial**

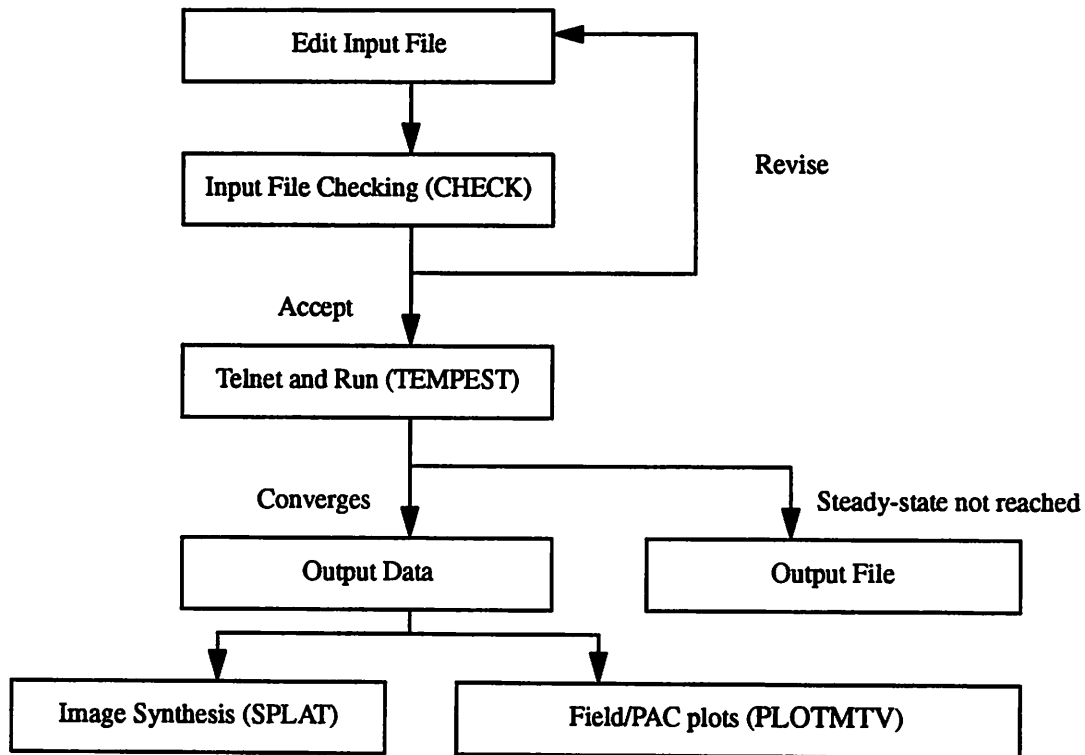
---

---

#### **3.1 Overview**

---

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



A typical simulation takes about 10 minutes to run on the CM-5. However, if a large amount of data (such as a contour plot of a large area) is requested, the simulation time can be as much as one hour.

---

#### **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 convention chrome mask is studied. The input file is located in the directory `$HOME/bin/TEMPEST/example/example1` and is named *example1.in*. The dimension of the simulation domain is  $4.0\text{ }\mu\text{m}$  by  $4.0\text{ }\mu\text{m}$  by  $1.0\text{ }\mu\text{m}$ . 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  $1.0\text{ }\mu\text{m}$  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 *example1.in* is shown as follows:

```
/*
input file name: example1.in
*/

/*
simulation window is 4 by 4 by 1 microns with 256 simulation nodes in the x-
and y-directions and 64 nodes in the z-direction.
*/
x_node 256
y_node 256
z_node 64
x_dim 4.0
y_dim 4.0
z_dim 1.0
periodic

/*
wavelength of 248 nm with 1000 mW/cm^2 of incident flux density normally
incident with the electric field linearly polarized in the x-direction.
*/
wavelength 0.248
intensity 1.0
inc_the 0.0
inc_phi 0.0
rel_ex0 1.0

/*
transmitted harmonics are requested. An inconsistency of 1% in the fields can
```

be tolerated, and the maximum number of wave cycles simulated is 300.

```
*/
transmitted
max_cycle 300
err_tol 0.01

/*
First block is air - index 1.0
*/
block 1
figure rectangle
0.0 4.0 0.0 4.0 0.0 1.0
index 1.0 0.0
end_block

/*
Second block is glass - index 1.5
*/
block 2
figure rectangle
0.0 4.0 0.0 4.0 0.6 1.0
index 1.5 0.0
end_block

/*
Third block is chromium.
*/
block 3
figure rectangle
0.0 4.0 0.0 4.0 0.52 0.6
index 2.5 2.0
end_block

/*
Fourth block is air again.
*/
block 4
figure rectangle
1.50 2.50 1.50 2.50 0.52 0.6
index 1.0 0.0
end_block

/*
Steady-state electric field plots.
*/
plot steady xyplane 0.00 e x position 0.0 4.0 0.0 4.0 0.5
example1.ex.xypln.000
plot steady xyplane 0.25 e x position 0.0 4.0 0.0 4.0 0.5
example1.ex.xypln.250
```

```
plot steady yzplane 0.00 e x position 0.0 4.0 0.0 1.0 2.0
example1.ex.yzpln.000
plot steady yzplane 0.25 e x position 0.0 4.0 0.0 1.0 2.0
example1.ex.yzpln.250
```

---

### 3.3 Checking the Input File

---

Since computation time on the CM is expensive, it is desirable to ensure that the format of the input file is correct before submitting it to be simulated. The routine CHECK checks an input file for possible errors. If the input file contains errors such as incorrect keywords, the program notifies the user of the mistake.

To check the input file *example1.in*, type the command:

```
ws% check example1.in
```

With this command, the following is displayed, indicating that there is no error in the input file *example1.in*:

```
C H E C K
ver3.0
94/07/14
```

```
Reading input file [example1.in]...
Done reading input file [example1.in].
```

```
No error detected. CHECK executed successfully.
```

---

### 3.4 Running Simulations

---

With no error messages from the checking routine CHECK, the input file is in the correct format and TEMPEST can be run. To execute the program TEMPEST, the user must be attached to a partition of the CM. A partition controls a certain number of nodes. The actual number of nodes each partition can control depends on the size of the CM. For instance, the CM-5 of the National Center for Supercomputing Applications (NCSA) at Urbana-Champaign, Illinois has 512 nodes (a 512-node machine) divided into units with 256, 128, 64, and 32 nodes. When one attaches to the CM through a partition manager, one tells the partition manager the number of nodes needed. The number of nodes requested is related to the problem size. In TEMPEST, the minimum number of nodes required is given by the product of the simulation nodes in the x-, y- and z-directions divided by the number 65536.

The first step to attach to a CM is to telnet or remote login to a partition manager of the CM. For example, to login to user "tempest" on the NCSA machine, type the following command:

```
ws% rlogin cm5.ncsa.uiuc.edu -l tempest
```

After logging into the partition manager, the user can request to attach to a partition by:

```
cm% jrun
```

The machine then asks several questions as follows:

```
Number of processors (32)? 64
Estimated cpu time (5min)? 15
Estimated memory (32M)? 1.5G
Job id is 94080.
[Type ~^B to disconnect, ~^C to kill. Waiting for resources.]
```

In this example, 64 nodes and 1.5 GBytes of memory are requested. The partition manager indicates that there is no free space on the CM-5 at the moment and the process has to wait. To check for the availability of the different partitions, use the command

```
cm% jstat
```

This command displays all the users who are currently using the CM-5:

```
USER JID MACH SERV PROCS TIME MEM STATUS COMMAND
tempest 94080 cm5 64 15:00 512M Que dwn /bin/csh
jbokor 94079 cm5 32 5:00i 32M Que dwn /bin/csh
blee 94071 cm5 cm5b 32 11:46? 53.4M Running trial4/gup4
afreeman 94066 cm5 cm5a 32 0:00? 0 Exiting /bin/csh
joannop 94046 cm5 cm5c 64 19:18? 590M Running sill11-7/sill11-dyn1
bernardo 94011 cm5 cm5d 128 98:55? 731M Running ammonia/lj/elmo_lj
fchen 93950 cm5 cm5a 32 59:44? 476M Running yeeinv2d/sfinv2d
sadayap 93117 cm5 cm5e 256 68:50? 550M Running kaushik/perms/datascript.256
```

The line which starts with “tempest 94080” (the first line) indicates that the process is sitting in the queue. When the queue finally becomes free, we can gain access to the connection machine.

```
[Connected to cm5e.ncsa.uiuc.edu /dev/tty2.]
*** Job csh (94080) procs 256, mem 1.5G, cpu 1hr, server cm5e:0
*** tempest /bin/csh started on Sun Jul 3 16:59:30 1994
```

We are now ready to run TEMPEST. There are two arguments for the program TEMPEST. The first argument is the input file name and is *example1.in* in this case, and *example1.out* is the name of the output file. The output file is the file to which information on the diffraction harmonics and simulation parameters is written.

```
[Connected to cm5e.ncsa.uiuc.edu /dev/tty2.]
*** Job csh (94080) procs 256, mem 1.5G, cpu 1hr, server cm5e:0
```

```
*** tempest /bin/csh started on Sun Jul 3 16:59:30 1994
cm5e:/nfs/cm5/scratch3/tempest [51]% tempest example1.in example1.out
```

```
TEMPEST
ver3.0
94/07/14
```

```
Start simulation...
```

```
Reading input file [example1.in]...
Done reading input file [example1.in].
```

```
Initializing parameters and variables...
```

```
Warning:
Minimum number of simulation nodes per wavelength for the topography is 6.35,
fewer than 15. Simulation result may be inaccurate.
```

```
Done initializing parameters and variables.
```

```
Start iteration...
Period #11:
1st try: 91.59% converged.
Period #12:
1st try: 93.15% converged.
Period #13:
1st try: 91.98% converged.
Period #14:
1st try: 95.50% converged.
Period #15:
1st try: 97.46% converged.
Period #16:
1st try: 99.22% converged.
Period #17:
1st try: 99.02% converged.
Period #18:
1st try: 98.63% converged.
Period #19:
1st try: 100.00% converged.
Period #20:
1st try: 100.00% converged.
2nd try: 98.83% converged.
Period #21:
1st try: 100.00% converged.
2nd try: 100.00% converged.
Period #22:
1st try: 100.00% converged.
2nd try: 100.00% converged.
3rd try: 100.00% converged.
Converges after 22 periods.
```

Done iteration.

Writing plot file [example1.ex.xypln.000]...  
Done writing.

Writing plot file [example1.ex.xypln.250]...  
Done writing.

Writing plot file [example1.ex.yzpln.000]...  
Done writing.

Writing plot file [example1.ex.yzpln.250]...  
Done writing.

Calculating diffraction efficiencies...  
Done calculating diffraction efficiencies.

Writing to output file [example1.out]...  
Done writing.

Simulation successful. Thank you.

When the program finishes execution, we need to detach from the CM. To do so, do the following:

```
cm5e:/nfs/cm5/scratch3/tempest [52]% exit
cm5e:/nfs/cm5/scratch3/tempest [53]% *** Job terminated at Sun Jul 3
17:16:41 1994
*** exitcode = 0000
*** procs 256, mem 1.56G, cpu 7.77min
***
```

Five data files are generated for this simulation:

1. *example1.out* (output file which contains information on the diffraction efficiencies)
2. *example1.ex.xypln.000* and *example1.ex.xypln.250* (electric field plots across an xy-plane in PLOTMTV format)
3. *example1.ex.yzpln.000* and *example1.ex.yzpln.250* (electric field plots across an yz-plane in PLOTMTV format)

These files can be transferred to be processed on a workstation by the file transfer program **ftp** or the remote copying command **rcp**.

### 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 as well. This information can be used to synthesize the optical image profile of the structure using the image synthesis program IMAGE2D which calls the simulation program SPLAT.

The output file generated, *example1.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

ver3.0

94/07/14

University of California, Berkeley

Miscellaneous Information:

Input file name: example1.in

Output file name: example1.out

There are 4 plot files:

1. Steady-state field example1.ex.xypln.000
2. Steady-state field example1.ex.xypln.250
3. Steady-state field example1.ex.yzpln.000
4. Steady-state field example1.ex.yzpln.250

Total time elapsed: 4.7165e+02 second

CM busy time: 60.59%

CM idle time: 39.41%

Process Parameters:

Freespace wavelength: +2.4800e-01 um

Incident theta: +0.0000e+00

Incident phi: +0.0000e+00

Refl abs theta 1: +0.0000e+00

Refl abs theta 2: +4.1345e-02

Tran abs theta 1: +0.0000e+00

Tran abs theta 2: +6.2040e-02



Incident relative Ex: +1.0000e+00  
Ey: +0.0000e+00  
Incident cosine Ex: +1.0000e+00  
Ey: +0.0000e+00  
Ez: +0.0000e+00  
Hx: +0.0000e+00  
Hy: -1.0000e+00  
Hz: -0.0000e+00  
Kx: +0.0000e+00  
Ky: +0.0000e+00  
Kz: -1.0000e+00  
Illumination intensity: +1.0000e+00 mW/cm2  
Mask file name: None

Topography:

The structure is assumed to be periodic.  
x dimension: 4.0000e+00 um  
y dimension: 4.0000e+00 um  
z dimension: 1.0000e+00 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 (+1.0000e+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 4.0000e+00  
yl yh 0.0000e+00 4.0000e+00  
zl zh 0.0000e+00 1.0000e+00

Block 2:

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 4.0000e+00  
yl yh 0.0000e+00 4.0000e+00  
zl zh 6.0000e-01 1.0000e+00

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 4.0000e+00
yl yh 0.0000e+00 4.0000e+00
zl zh 5.2000e-01 6.0000e-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 1.5000e+00 2.5000e+00
yl yh 1.5000e+00 2.5000e+00
zl zh 5.2000e-01 6.0000e-01
```

**Computational parameters:**

```
Distance between nodes (dx): 1.5625e-02 um
Simulation grid: 256 x 256 x 64
Minimum nodes per wavelength: 6.35
Time increment (dt): 2.9544e-17 second
Iterations per period (wave cycle): 28
```

**Convergence information:**

```
Wave cycles before convergence check: 10
Maximum iteration wave cycles: 300
Error tolerance: 1.00%
Number of periods for convergence: 22
```

**Transmitted diffraction efficiencies:**

```
Number of diffracted orders: 16 16
```

```
Real Imag
```

```
...
order[ +0 ][ -3 ]: +5.037091e-05 -1.923219e-02
-2.716657e-05 -1.102946e-05
-2.739282e-05 -1.126659e-05
-8.187038e-06 +1.875915e-02
order[ +0 ][ -2 ]: +1.830793e-03 +3.778544e-02
+4.256049e-05 +2.094013e-05
+4.297282e-05 +2.133218e-05
-1.989200e-03 -3.747990e-02
order[ +0 ][ -1 ]: -5.006125e-03 -5.188354e-02
-3.576368e-05 -1.723806e-05
```

```
-3.631501e-05 -1.741823e-05
+5.449582e-03 +5.185812e-02
order[ +0 ][ +0 ]: +7.808192e-03 +5.096931e-02
+9.865538e-06 +5.952744e-06
+1.003706e-05 +6.072602e-06
-7.720337e-03 -5.192857e-02
order[ +0 ][ +1 ]: -4.367713e-03 -5.192599e-02
+2.185255e-05 +8.608084e-06
+2.218210e-05 +8.654256e-06
+4.810850e-03 +5.190511e-02
order[ +0 ][ +2 ]: +9.023458e-04 +3.779706e-02
-4.188506e-05 -2.162569e-05
-4.227937e-05 -2.203745e-05
-1.067477e-03 -3.749463e-02
order[ +0 ][ +3 ]: +7.577330e-04 -1.920048e-02
+3.816728e-05 +2.329560e-05
+3.857693e-05 +2.368192e-05
-6.985012e-04 +1.872935e-02
...
```

Transmitted energy: 5.21%

Total refl and tran energy: 30.58%

---

### **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.

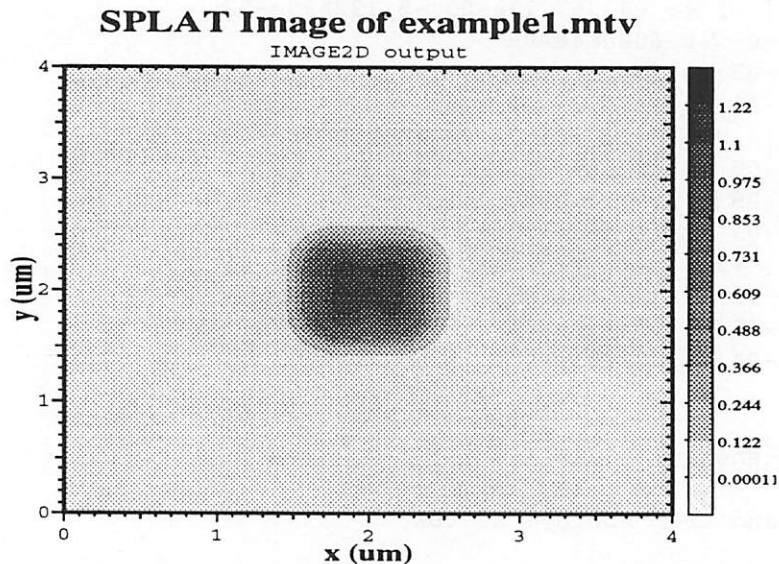
With these assumptions, the intensity at every point in space can be calculated by evaluating the transmission cross coefficients<sup>1</sup>. This approach can be used to model bright-field or dark-field optical systems, arbitrary lens aberrations as well as defocus effects.

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% image2d example1.out -na 0.5 -sg 0.5 -m 1 -p
```

where *example1.out* is the output file of TEMPEST. The command **image2d** is described in detail in Section 4.4.1.

With the above command, the image in Figure 1 is displayed on the screen. The data points of the image is written to the file *example1.mtv*.



**Figure 1** The optical signal created by a  $1\text{ }\mu\text{m}$  by  $1\text{ }\mu\text{m}$  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

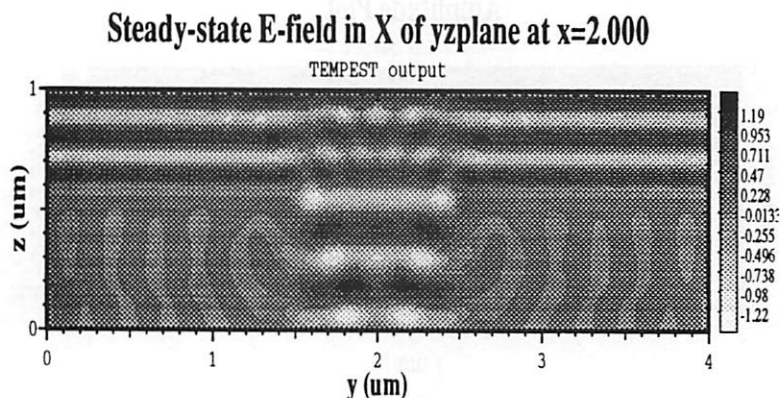
```
ws% plotmtv file_name
```

For example, to display the data file *example1.ex.yzpln.000*, type

```
ws% plotmtv example1.ex.yzpln.000
```

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

From the instantaneous values of the fields, it is possible to calculate the amplitude and phase using the routines AMPHA1D and AMPHA2D. For example, from the electric field data *example1.ex.yzpln.000* and *example1.ex.yzpln.250*, AMPHA2D can be used to find the transmitted electric field amplitude:



**Figure 2** The instantaneous electric field in the x-direction across an yz-plane of the simulation domain. The fields show that the electric field is diffracted as it exits the chromium opening.

```
ws% ampha2d example1.ex.yzpln.000 example1.ex.yzpln.250
example1.ex.yzpln.amp example1.ex.yzpln.pha
```

Sometimes raw data files such as *example1.ex.yzpln.amp* are too large. The routine **reduce2d** reduces the size of the data file by taking one out of a few data points from the raw data file. For example, the data file *example1.ex.yzpln.amp* can be processed as follows:

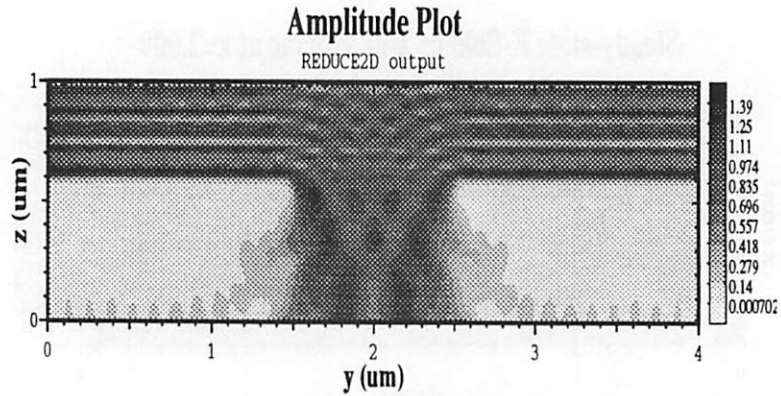
```
ws% reduce2d example1.ex.yzpln.amp example1.ex.yzpln.amp.red 2 2
```

The file *example1.ex.yzpln.amp.red* contains one datum out of two data points in *example1.ex.yzpln.amp* in both the x- and y-directions.

A plot of the file *example1.ex.yzpln.amp.red* using PLOTMTV is shown in Figure 3.

### 3.6 Example 2: Photoresist Exposure

In the second example, the image synthesized from the first example is used to expose a layer of photoresist deposited over a stepped substrate. The volume of the simulation domain is  $4\text{ }\mu\text{m}$  by  $4\text{ }\mu\text{m}$  by  $1\text{ }\mu\text{m}$ . The ABC parameters of the photoresist is (0.51, 0.031, 0.013).  $100\text{ mJ/cm}^2$  is deposited onto the photoresist. Bleaching of the photoresist is considered by breaking up the total exposure dose into five steps.



**Figure 3** Electric field amplitude transmitted through a chromium mask. The data file is generated by the routine AMPHA2D.

### 3.6.1 Input File

The input file named *example2.in* is shown below.

```

/*
input file name: example2.in
*/

/*
volume of simulation is 4 by 4 by 1 microns with 256 nodes in the x- and the
y-directions, and 64 nodes in the z-direction.
*/
x_node 256
y_node 256
z_node 64
x_dim 4.0
y_dim 4.0
z_dim 1.0
periodic

/*
the incident radiation is monochromatic at 248 nm. The incident field
amplitude profile is taken from the file named "example1.mtv"
*/
wavelength 0.248
intensity 1.0
mask example1.mtv

/*
no diffracted efficiencies are requested.
*/
no_de

```

```
max_cycle 300
err_tol 0.01

/*
first block is air
*/
block 1
figure rectangle
0.0 4.0 0.0 4.0 0.0 1.0
index 1.0 0.0
end_block

/*
second block is photoresist
*/
block 2
figure rectangle
0.0 4.0 0.0 4.0 0.15 0.75
index 1.68 0.02
resist 1.0 0.51 0.031 0.013 100.0 5
end_block

/*
third block is silicon
*/
block 3
figure rectangle
0.0 4.0 0.0 4.0 0.00 0.3
index 4.0 2.0
end_block

/*
fourth block is resist
*/
block 4
figure rectangle
1.25 2.75 1.25 2.75 0.15 0.3
index 1.68 0.02
resist 1.0 0.51 0.031 0.013 100.0 5
end_block

/*
PAC plots are requested
*/
plot pac xyplane position 0.0 2.0 0.0 2.0 0.4 example2.pac.xypln
plot pac yzplane position 0.0 2.0 0.15 0.75 2.0 example2.pac.yzpln
```

### 3.6.2 Running Simulation

---

Assuming that the routine CHECK has been applied to the input file and the input file is in the correct format, TEMPEST simulation can be run on the input file *example2.in*.

**ws% rlogin cm5.ncsa.uiuc.edu -l tempest**

**cm% jrun**

```
cm5:~ [34]% jrun
Number of processors (32)? 64
Estimated cpu time (5min)? 15
Estimated memory (64M)? 1.5G
Job id is 94252.
[Type ~^B to disconnect, ~^C to kill.]
[Connected to cm5e.ncsa.uiuc.edu /dev/ttypl.]
*** Job csh_94252 (94252) procs 256, mem 1.5G, cpu 15min, server cm5e:0
*** tempest /bin/csh started on Wed Jul 6 19:03:51 1994
cm5e:/nfs/cm5/scratch3/tempest [51]% tempest example2.in example2.out
```

```
T E M P E S T
ver3.0
94/07/14
```

Start simulation...

Reading input file [example2.in]...  
Done reading input file [example2.in].

Initializing parameters and variables...

Warning:

Minimum number of simulation nodes per wavelength for the topography is 3.97,  
fewer than 15. Simulation result may be inaccurate.

Reading image information from [example1.mtv]...  
Done reading image information.

Done initializing parameters and variables.

Iterating for energy step 1...  
Start iteration...

Period #17:  
1st try: 100.00% converged.

Period #18:  
1st try: 100.00% converged.  
2nd try: 100.00% converged.

Period #19:  
1st try: 100.00% converged.  
2nd try: 100.00% converged.



3rd try: 100.00% converged.  
Converges after 19 periods.  
Done iteration.

Updating parameters in the resist layer...  
Done updating parameters.

Total exposure time: 1.209e+01 seconds.  
Total dose: 1.209e+01 mW/cm2.

Iterating for energy step 2...  
Start iteration...  
Period #1:  
1st try: 100.00% converged.  
Period #2:  
1st try: 100.00% converged.  
2nd try: 100.00% converged.  
Period #3:  
1st try: 100.00% converged.  
2nd try: 100.00% converged.  
3rd try: 100.00% converged.  
Converges after 3 periods.  
Done iteration.

Updating parameters in the resist layer...  
Done updating parameters.

Total exposure time: 2.645e+01 seconds.  
Total dose: 2.645e+01 mW/cm2.

Iterating for energy step 3...  
Start iteration...  
Period #1:  
1st try: 100.00% converged.  
Period #2:  
1st try: 100.00% converged.  
2nd try: 100.00% converged.  
Period #3:  
1st try: 100.00% converged.  
2nd try: 100.00% converged.  
3rd try: 100.00% converged.  
Converges after 3 periods.  
Done iteration.

Updating parameters in the resist layer...  
Done updating parameters.

Total exposure time: 4.412e+01 seconds.  
Total dose: 4.412e+01 mW/cm2.

Iterating for energy step 4...

Start iteration...

Period #1:

1st try: 100.00% converged.

Period #2:

1st try: 100.00% converged.

2nd try: 100.00% converged.

Period #3:

1st try: 100.00% converged.

2nd try: 100.00% converged.

3rd try: 100.00% converged.

Converges after 3 periods.

Done iteration.

Updating parameters in the resist layer...

Done updating parameters.

Total exposure time: 6.709e+01 seconds.

Total dose: 6.709e+01 mW/cm2.

Iterating for energy step 5...

Start iteration...

Period #1:

1st try: 100.00% converged.

Period #2:

1st try: 100.00% converged.

2nd try: 100.00% converged.

Period #3:

1st try: 100.00% converged.

2nd try: 100.00% converged.

3rd try: 100.00% converged.

Converges after 3 periods.

Done iteration.

Updating parameters in the resist layer...

Done updating parameters.

Total exposure time: 1.000e+02 seconds.

Total dose: 1.000e+02 mW/cm2.

Writing plot file [example2.pac.xypln]...

Done writing.

Writing plot file [example2.pac.yzpln]...

Done writing.

Writing to output file [example2.out]...

Done writing.

Simulation successful. Thank you.

```
cm5e:/nfs/cm5/scratch3/tempest [52]% exit
cm5e:/nfs/cm5/scratch3/tempest [53]% *** Job terminated at Sun Jul 3
17:16:41 1994
*** exitcode = 0000
*** procs 256, mem 1.56G, cpu 7.77min
***
```

Three output files are produced in this simulation:

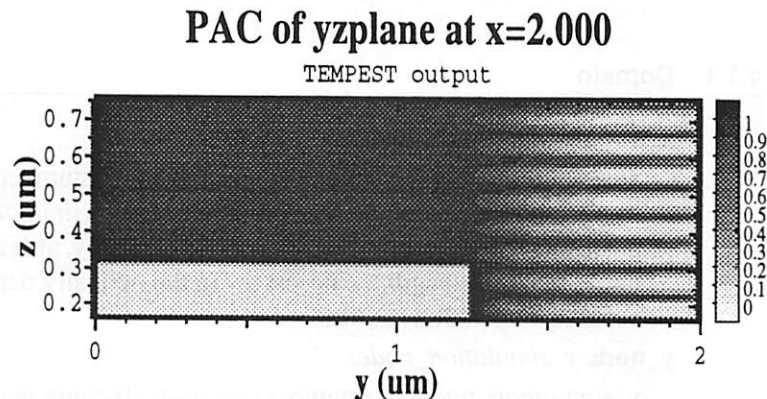
1. *example2.out* (output file which contains information on the simulation run)
2. *example2.pac.xypln* (contours of the PAC across an xy-plane)
3. *example2.pac.yzpln* (contours of the PAC across an yz-plane)

The output file *example2.out* is in a form similar to that of the first example shown in Section 3.5.1, and hence it will not be shown here.

### 3.6.3 Output Data

To display the PAC contours, again using the plotting package PLOTMTV. For instance, to display Figure 4 on the screen, type

```
ws% plotmtv example2.pac.yzpln
```



**Figure 4** PAC contours calculated by TEMPEST. The photoresist is exposed by the image synthesized in the first example.

---

## 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. *x\_simulation\_nodes* must be a positive number which is a power of 2. 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. *y\_simulation\_nodes* must be a positive number which is a power of 2.

The ratio of *y\_simulation\_nodes* to *x\_simulation\_nodes* must be the same as the ratio between the physical dimensions in y and x.

##### **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. *z\_simulation\_nodes* must be a positive number which is a power of 2. The product of *x\_simulation\_nodes*, *y\_simulation\_nodes*, and *z\_simulation\_nodes* cannot exceed the product of the number of CM-5 nodes and 65536. For example, if one is attached to 256

nodes, then the product (*x\_simulation\_nodes* times *y\_simulation\_nodes* times *z\_simulation\_nodes*) must be fewer than or equal to (256 times 65536).

#### **periodic**

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

#### **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 CM to which the user has access.

#### **isolated**

keyword **isolated** specifies that the structure under study is an isolated structure. The default structure type is **periodic**.

#### **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), and **no\_de** should also be specified.

#### **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), and **no\_de** should be specified.

---

### 4.1.2 Illumination

---

#### **wavelength lambda**

*lambda* is the free space wavelength of the incident illumination in micro-meter. 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  $\hat{E}$ ,  $\hat{H}$ , and  $\hat{k}$  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 IMAGE2D.

---

#### 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 120.

**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 (*y1*, *z1*), (*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 (*z1*, *x1*), (*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 (*x1*, *y1*), (*x2*, *y2*), and (*x3*, *y3*). The wedge is oriented along the z-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 post-processor IMAGE2D 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.

**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:

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

**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 six 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**, **h**, or **s**, specifying the electric field, the magnetic field, or the Poynting vector. The second entry can be one of the following: **x**, **y**, **z**, or **t**. **x**, **y**, or **z** specifies the direction of the field, and **t** specifies that the total field (magnitude) is requested.

*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 micrometers. **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 jrun

---

Selected sections of the manual pages of **jrun** are shown below. Interested users can read the on-line manual by typing

**cm% man jrun**

#### NAME

**jrun** - run a job interactively (DJM)

#### SYNOPSIS

**jrun** [file] [options]

**jrun** [options] -- file [arguments]

#### DESCRIPTION

The **jrun** command runs a job interactively. The output is directed to your terminal and the input is obtained from your keyboard. To disconnect your terminal from the job, type a tilde (~) followed by control-B (^B). After you disconnect, the input and output will be redirected to disk files or /dev/null as indicated by the -stdin and -stdout options. The job can later be reconnected to the terminal using **jconnect**.

If no program is specified on the command line, your login shell is invoked and an interactive login session is created. The options to **jrun** are a superset of those provided by **jsub**. Options may be specified on the command line or they can appear as embedded comment directives within the job's shell script. See **jsub**.

At minimum you must specify the following options: **-nproc**, **-cputime**, and **-memory**. These are required by DJM in order to schedule jobs effectively. When disconnected, output will be sent to a file with the same name as the job file with the suffix .onnn appended, where nnn is the job id. If the EMACS environment variable is set, **jrun** assumes it is running inside of an emacs shell-mode buffer.

#### OPTIONS

The options unique to **jrun** are defined below. See **jsub** for a definition of the remaining options.

##### **-capture**

Always capture output to the -stdout file, even when the job is connected to the terminal. Normally output is captured only when the job is disconnected.

##### **-echo**

Turn on echo. The session will echo characters typed at the terminal. This is the default.

##### **-emacs**

Turn on emacs mode. This option is useful when using **jrun** inside of an emacs shell-mode buffer. It disables character echo and disables control-S

and control-Q processing. This option overrides the EMACS environment variable.

**-noecho**

The session should not echo characters typed at the terminal.

**-noemacs**

Turn off emacs mode. This option overrides the EMACS environment variable.

**-eightbit**

Pass 8-bit characters through unchanged. By default the uppermost bit of each character is stripped off to create 7-bit logical characters.

---

#### **4.2.2 jsub**

---

Selected sections of the manual pages of **jsub** are shown below. Interested users can read the on-line manual by typing

**cm% man jsub**

**NAME**

**jsub** - submit a job for background execution (DJM)

**SYNOPSIS**

**jsub** [file] [options]  
**jsub** [options] -- file arguments]

**DESCRIPTION**

The **jsub** command submits a file containing a program or a shell script for execution on a machine. The job will be executed in the same directory in which the job was submitted. The job will be run in the background and all output will be directed to a file. The job can be connected to a terminal if the **-tty** option is set. See **jconnect**.

**jsub** is upward compatible with the command **qsub** in the Network Queuing System (NQS).

Options may be specified on the command line or they can appear as embedded comment directives within the job's shell script. Such directives consist of a shell comment character (**#**) followed by a special prefix: either **J**, **JSUB**, **QSUB**, **PBS**, or **@\$**. The first line of the script can begin with **#!** to specify a shell to interpret the script. Example: **#!/bin/sh**. If omitted, the login shell is used.

An example is shown below of using embedded comment directives in a job script.

```
#!/bin/sh the name of the shell
#JSUB -nproc 8K
#JSUB -cpu 60min -mem 128mb
#JSUB
a.out # run my program
```

Directives are evaluated up to the first non-blank, non-comment line or up to the first directive with no - present.

The output will be sent to a file with the same name as the job with the suffix .onnn appended, where nnn is the job id. By default, the standard output and the standard error are combined into a single stream and sent to this file. You can redirect the standard error to a different file with the -stderr option. This option will not work if -tty is set because there is no way to disambiguate output sent to a terminal device.

At minimum you must specify the following options: **-nproc**, **-cputime**, and **-memory**. These are required by DJM in order to schedule jobs effectively. It is in your best interest to estimate these resource values as accurately as possible with respect to your job's actual needs. If you understate your job's resource requirements, your job may be terminated by DJM if other jobs are waiting to run. On the other hand, you should not overestimate your resource requirements as that may cause your job's execution to be delayed.

These estimates are treated as "soft" limits when your job runs. Your job is not automatically killed if they are exceeded. The chance of getting killed depends on how egregious the overrun is and how heavily loaded the machine is.

The actual resource usage is written to the end of the output when your job finishes. You can use this information to supply improved usage estimates the next time you run your job.

#### **OPTIONS**

All option names may be abbreviated to uniqueness.

##### **-machine mach**

Execute the job on machine mach. This is typically the name of a Connection Machine (Reg.) system. By default the machine name is obtained from the environment variable DJM\_MACHINE.

##### **-cputime n**

Request n units of CPU time. The suffix secs indicates seconds, min indicates minutes, hr indicates hours, and days indicates days. The default is min. Suffixes may be abbreviated. Example: **-cpu 10h**. The amount of time is limited based on per-queue and global maximums. See jstat limits.

##### **-memory n**

Request n units of memory. The suffix kb indicates kilobytes, kw indicates kilowords (8192 bytes), mb indicates megabytes, mw indicates megawords, gb indicates gigabytes, and gw indicates gigawords. The default is megabytes. A single letter abbreviation indicates a byte (not word) unit. Example: **-mem 2g**. The amount of memory is limited based on per-queue and global maximums. Note that the amount represents total memory, not memory per processor.

##### **-nproc n**

Request at least n processors. The suffix k indicates 1,024 processors. Note that this represents a minimum requirement: your job may actually be run on

a partition with more processors than you requested if the partition is available. If you require exactly *n* processors, add the **-exact\_proc** option.

**-dedicate**

Request dedicated access to the partition. Your job will have the right to access of all resources (memory, cpu, nproc) of the partition during its execution. The **-memory** option is ignored, as you own all of the memory by definition. The job will only be run when there exists a partition that is enabled for dedicated access. Typically the operator will create one such partition encompassing the entire machine overnight. Dedicated partitions have special scheduling rules. See the DJM Administration Guide for more details. (NOTE: Do not confuse **-dedicated** with **-notimeshared**. The latter merely runs a job on a sequencer that does not have a timesharing daemon.)

**-after time**

Defer execution of the job until after the indicated time (or date). If the parameter contains embedded spaces, it must be enclosed in quotes, for example: **-after** "July 14 12:30pm".

**-hard\_cputime n**

Specify a hard limit for cpu time. If your job exceeds this limit, it is killed immediately. This is useful for preventing a runaway job from consuming all of a user's account allocation.

**-help**

Display a summary of all options along with a one line description of each option.

**EXAMPLE**

**cm% cat myjob**

```
#!/bin/sh
#
# This script shows an example of embedded DJM comment directives
#JSUB -procs 8K
#JSUB -cpu 60min
#JSUB -mem 80M
#
a.out
```

**cm% jsub myjob -mach cm200**

Job submitted successfully. Job id is 57.

(one hour later)

**cm% cat myjob.o57**

```
*** Distributed Job Manager 1.0, CM-200 (pa) AHPCRC / Minn. Supercomputer Ctr
*** job57 procs 8192, mem 80M, cpu 60min, seq 2, server cb:0
*** alan /bin/sh myjob started on Fri May 22 14:18:01 1992
Attached to {CM}* timesharing on CM-MSD, a CM200 on interface 0
Attached to 8192 CM200 processors on sequencer 2, microcode version 6104
1024K of memory with 64 bit floating point. Paris safety is on.
+ a.out
Working...
```

I'm done.  
Detached from (CM)\* Timesharing.  
\*\*\* Job terminated at Fri May 22 15:20:15 1992  
\*\*\* exitcode = 0000  
\*\*\* procs 8192, mem 43M, cpu 52min  
\*\*\*  
\*\*\* You requested 80M of memory but only used 43M.  
\*\*\* The next time you run this job, you should request  
\*\*\* less memory. This will let your job run earlier  
\*\*\* and you will get your results sooner.

#### NOTES

There are per-queue and per-user limits on the number of jobs that can be submitted at the same time.

A **jsub** job with the **-tty** option is indistinguishable from a **jrun** job.

The Posix Batch System (IEEE Std 1003.15) changed the way options are parsed, requiring all options to consist of a single letter. DJM, on the other hand, has many more arguments than available letters in the alphabet. This means that options which can be grouped together in PBS (for example, **-zV**) must be broken apart in DJM (**-z -V**).

---

### 4.2.3 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 diffraction harmonics is written.

**cm% tempest** *input\_file output\_file*

---

## 4.3 Pre-TEMPEST Workstation Commands

---

---

### 4.3.1 CHECK

---

Since computation time on the CM is expensive, it is desirable to ensure that the format of the input file is correct before submitting the input file for simulation. The CHECK routine checks for possible errors in the input file. The program CHECK is invoked in the following way:

**ws% check** *input\_file*

where *input\_file* is the input file to TEMPEST. If the input file *input\_file* contains mistakes such as incorrect keywords, CHECK notifies the user of the mistake.



---

## 4.4 Post-TEMPEST Workstation Commands ---

---

### 4.4.1 IMAGE2D ---

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 program IMAGE2D, which calls SPLAT to calculate the aerial images. Since TEMPEST assumes coherent illumination, partial coherence in the IMAGE routine 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.

IMAGE2D is invoked with the following command:

**image** *TEMPEST\_output\_file* [-na *NA*] [-sg *sigma*] [-f *defocus*] [-m *magnification*] [-x *x\_displace*] [-y *y\_displace*] [-t *x\_pts y\_pts*] [-p]

*TEMPEST\_output\_file* is any output file generated from TEMPEST which contains information on the diffraction efficiencies. Optical image profiles at the defocus level specified by *defocus* is calculated and displayed on the screen if the [-p] flag is specified. The image data is written to the file *TEMPEST\_output\_file.mtv* in PLOTMTV format. If the user is working on a terminal with no graphical facility, the [-p] flag should not be specified. The characteristics of the optical system are described by the flags [-na], [-sg], and [-m]. *NA* is the numerical aperture of the optical system; *sigma* is the partial coherence factor of the system; *magnification* is the magnification factor.

Based on the parameters specified on the input line, IMAGE2D creates a SPLAT input file called *TEMPEST\_output\_file.spl*. The user can modify this SPLAT input file to model different optical system characteristics such as modified illumination, pupil filtering, lens aberrations, and/or thin-film effects. Please refer to the SPLAT Users' Guide<sup>3</sup> for more information.

**-na** *NA*

numerical aperture of the optical system. The default is 0.5

**-sg** *sigma*

sigma is the partial coherence factor. The default is 0.5

**-f defocus**

defocus is the amount of defocus in micro-meter. The default is 0.0

**-m magnification**

*magnification* is the magnification factor of the mask. For example, **-m 4** means that it is a 4X reduction system. The default is 5.

**-x *x\_displace***

displaces the calculated image by *x\_displace*  $\mu\text{m}$  in the x-direction. The option is used to facilitate the displaying of the image.

**-y *y\_displace***

displaces the calculated image by *y\_displace*  $\mu\text{m}$  in the y-direction. The option is used to facilitate the displaying of the image.

**-t *x\_pts y\_pts***

the optical image data is written to the file *TEMPEST\_output\_file.mtv* with *x\_pts* data points in the x-direction and *y\_pts* data points in the y-direction.

**-p**

displays the calculated image on the terminal.

---

**4.4.2 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.

---

**4.4.3 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.

---

#### **4.4.4 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*.

---

#### **4.4.5 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.

---

## 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 (which should not be any), try adding the line "**dtfrac 0.5**" to the TEMPEST input file.

---

**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 *SPLAT v5.0 Users' Guide*, ERL, University of California, Berkeley, 1994.
- 4 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.
- 5 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.
- 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 *CM5 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 42

**C**

**check**, input file checking routine 40

**F**

**factor1d/2d**, scaling programs for files in PLOTMTV format 43

**I**

**image2d**, image synthesis routine which calls SPLAT 41

**J**

**jrun**, run a CM job interactively 36

**jsub**, run a batch CM job 37

**P**

**plotmtv**, plotter 42

**R**

**reduce2d**, routine which converts large data files to smaller ones by discarding data points 43

**T**

**tempest**, electromagnetic fields solver 40

---

## Appendix B Connection Machine Facilities

---

The following list of connection machine facilities was obtained from David Ray (ray@think.com) of the Customer Support Group, Thinking Machines Corporation.

### **B.1 Thinking Machines Corporation - Connection Machine Network Server (CMNS) Pilot Facility**

Address: CMNS Facilities Manager  
Thinking Machines Corp.  
245 First Street  
Cambridge, MA 02142-1214  
Phone: (617)876-1111 ext. 2121  
Electronic Mail: cmns-manager@think.com  
uucp: ames!think!cmns.manager

### **B.2 NASA Ames Research Center**

Address: Mail Stop T045-1  
Moffett Field, CA 94035  
Phone: Bruce Blaylock (415)604-4400  
Dave Gambrel (415)604-4504

### **B.3 National Center for Supercomputing Applications (NCSA)**

Address: Metacomputer Allocations  
National Center for Supercomputing Applications  
257 Computing Applications Building  
605 E. Springfield Ave.  
Champaign, IL 61820  
Contact: allocations (217)244-0635 allocations@ncsa.uiuc.edu

### **B.4 Northeast Parallel Architectures Center (NPAC)**

Address: NPAC - Syracuse University  
111 College Place  
Science & Technology Bldg.  
Machine Room 1-213  
Syracuse, NY 13244-1240  
Phone: (315)443-1722  
Fax: (315)443-1973  
Contact: Don Hewitt (315)443-2086 don@nova.npac.syr.edu



**B.5 Naval Research Laboratory (NRL) [Code 51-30]**

**Address:** 4555 Overlook Ave., SW  
Washington, DC 20375-5000

**Phone:** (202)404-7020 or (202)404-7029

**Contacts:** Henry Dardy (202)404-7028 dardy@cmvax.nrl.navy.mil  
CM-Room (202)767-9044 or (202)767-9045