

033279-4-T

in file

USERS MANUAL FOR JETSWEEP

Hristos Anastassiu
Mark Casciato
Dan Ross
John Volakis

Wright Laboratories/AARA
Wright Patterson AFB
OHIO 45433-7001

DEMACO
Suite 303
100 Trade Center
Champaign, IL 61820



February 1996

THE UNIVERSITY OF MICHIGAN

Radiation Laboratory
Department of Electrical Engineering
and Computer Science
Ann Arbor, Michigan 48109-2122
USA

Users Manual for Jetsweep

Hristos T. Anastassiou, Daniel C. Ross, Mark D. Casciato,
and John L. Volakis

Radiation Laboratory
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor MI 48109-2122

February 3, 1996

1 Introduction

Jetsweep is a FORTRAN77 code which uses mode matching techniques to obtain the scattered field from jet engine cavities with straight and a certain class of curved blade sets. The code is separated into 3 parts. The first part is parallelized for the Intel Paragon and computes modal inner products. The second part of the code is serial and computes scattering matrices which will be used by the third part of the code (serial) to obtain the Radar Cross Section (RCS). The code is set up to run multiple aspects at multiple frequencies.

Section 1 of this manual will give a brief description of the mode matching and the parallelization methods used in Jetsweep. Section 2 describes the code operation. Section 3 presents Jetsweep examples including narrow band (NB, single frequency) plots as well as Synthetic Aperture Radar (SAR) images. Finally, section 4 gives the installation instructions along with a list of files contained in Jetsweep.

2 Mode Matching and Parallelization

This section provides a brief description of the mode matching techniques implemented in this code along with the employed parallelization techniques used. For more detailed descriptions of both the users are referred to [1]-[2].

Mode matching is an analytical technique used for cavity analysis. Referring to the geometry of Figure 1 [1], a circular cavity with a simple hub termination is first divided into 2 regions with separable geometries which can be analyzed analytically.

Cavity fields in each region are separated into incoming and outgoing components which are expanded into a series of transverse field modes with unknown weighting coefficients. The unknown coefficients are found by enforcing the tangential boundary conditions at the interface of the two regions and the back wall of the cavity. Once the coefficients are determined a scattering matrix is obtained and is subsequently used to extract the scattered fields. To do so it is first necessary to propagate the fields down the inlet circular tube and then employ aperture integration at the end of the tube.

In order to parallelize an existing serial code a profile must first be done as to which parts of the code consume large portions of time and memory [2]. By identifying these portions of the code and parallelizing the most time intensive sections of the code a significantly faster code can be produced with a minimal amount of labor. The profile of the mode matching code showed that over 90% of the run time was spent in matrix fill operations and that parallelizing this part of the code would be the most beneficial in reducing run times. The elements of these matrices are the inner products of different mode combinations at the interface between regions [2]. In order to keep memory demands at a minimum, all processors contribute to each matrix element as opposed to each processor computing a separate element. Jetsweep1 is the parallelized section of Jetsweep which computes these modal inner products. It was found that the remaining 10% of the time is used performing matrix inversion and matrix-matrix products [2], along with the evaluation of the far fields which require matrix-vector products. The large number of matrix-matrix products required would have made the evaluation of many blades impractical. By using the properties of overlapping modal and geometric symmetry, the problem was reduced to a manageable state for any number of blades [1]. Even with this significant improvement, however, the matrix inversion and matrix-matrix products require a large amount of inner-processor communication which will dominate the code run time as the number of processors are increased. As more processors are added the communication time will increase at a much greater rate than the gain in time from the added processors. Eventually, a point is reached where the parallel code will run slower than the serial version. For this reason, this part of the code was kept serial and is implemented in Jetsweep2. FFcyl uses the scattering matrices generated by Jetsweep2 to calculate far field scattering from the engine. While FFcyl is dominated by matrix-vector products and would benefit by parallelization, the computation of the far fields by a serial machine is a relatively fast operation and FFcyl is implemented as a serial code at this time.

3 Code Operation

Operation of the three Jetsweep codes is very simple. For Jetsweep1 and Jetsweep2 the input file 'file'.geo must be modified with appropriate input

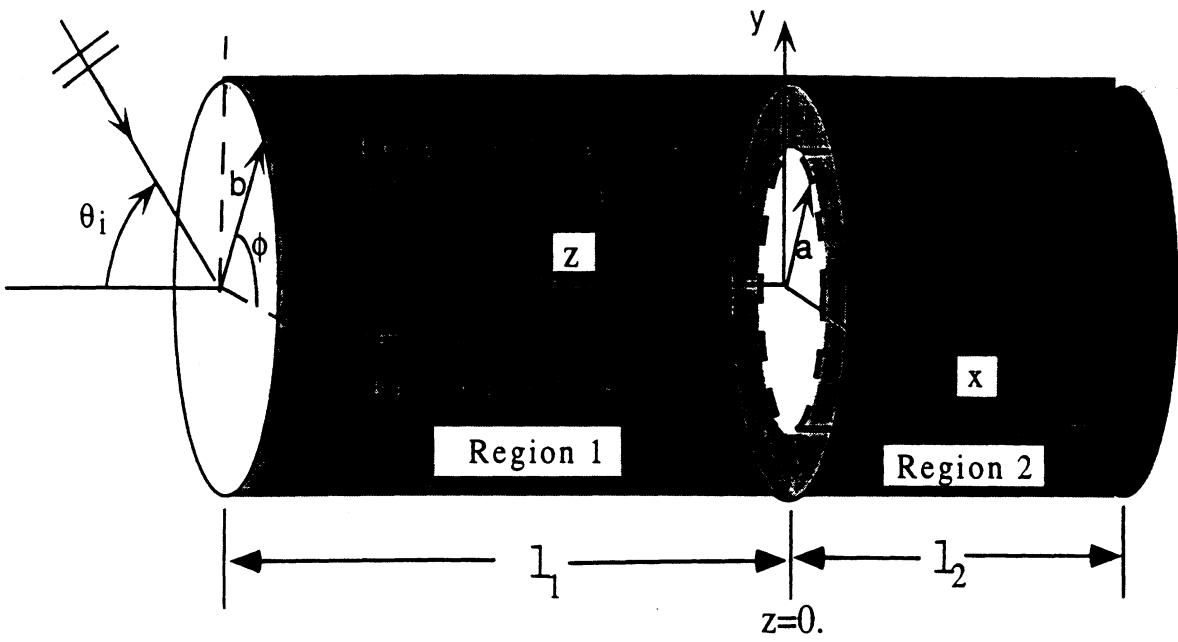


Figure 1: Engine Inlet Terminated by a Cylindrical Hub

parameters. An example with a description of each input is shown below:
'file'.geo - Input File:

```
1      ! 1: straight 2: curved (blades)
2      ! plot interface fields (1:yes 2:no), Usually left at 2
2000   ! fstart (MHz), Frequency start
18000  ! fend (MHz), Frequency stop
50     ! delta f (MHz), Frequency step size
15.0   ! outer radius (cm.), Inner radius of cavity
7.5    ! inner radius (cm.), Outer radius of blades
20.0   ! inlet length from mouth to blades (cm)
10.0   ! blade length (cm)
8      ! number of blades
40.0   ! groove width - air (degrees), Angle between blades
2.5    ! first groove position (degrees), Measured from center of blade
0.25   ! far field angle step- theta (degrees)
1      ! rim diffraction 1:on 2:off
2      ! scattering matrix? (1:yes 2:no), Usually left at 2
30.0   ! total twist angle (degrees), For curved blades only
1D12   ! very large number (see code)
```

The executable jetsweep1, runs on an Intel Paragon parallel platform. Jetsweep2 runs on a Sun platform. These are script files which manage the compiling and execution of the codes through frequency sweeps. For each frequency, these scripts set up appropriate parameters and rewrite a fortran include file. The main fsweep1.f and fsweep2.f routines are then automatically recompiled and run for each frequency. Before running jetsweep1 the parameter set n='number of processors' in the jetsweep1 script must be set to the appropriate number of processors. Usually this is set as high as possible (the maximum processors available). To run jetsweep1 on the Paragon simply type:

```
jetsweep1 'file'
```

where 'file' is the name of the 'file'.geo input file. The output files, 'file'_mat1_'fMHz' which are in unformatted Paragon binary must be stripped of headers and rewritten before porting to a Sun platform for jetsweep2. This is done by the script jetcvtpara which performs a similar function to the jetsweep1 and jetsweep2 scripts. To execute type:

```
jetcvtpara 'file'
```

The output are files are 'file'_mat1_'fMHz'.sun which are now ported to a Sun platform for jetsweep2 execution.

On the Sun platform the 'file'_mat1_'fMHz'.sun files must now be converted to a Sun compatible binary. This is done by the script jetcvtsun, again with a function similar to the previous scripts. To run type:

```
jetcvtsun 'file'
```

where 'file' is the name of the same 'file'.geo input file. The output files will again be 'file'_mat1_'fMHz'. To generate the scattering matrices with jetsweep2 on the Sun platform type:

```
jetsweep2 'file'
```

The output ascii files are now designated 'file'_mat2_'fMHz'. These are the scattering matrices in 2D sparse matrix format. These files are now ported to an SGI workstation for final far field calculations.

Before far fields are computed on the SGI platform the 'file'_mat2_'fMHz' files must be converted to an 8D sparse matrix format for final processing. This is done by executing the script sctmatcvt on the SGI. Type:

```
sctmatcvt 'file'
```

The output files are designated 'file'_'fMHz'.scatmat and are now ready for final processing with the FFcyl routines on the SGI. The manager for the final far field calculations is the fortran routine FF.f which executes the main FFcyl.f routine through frequency. Note that if a shorted guide with no blades is desired the logical variable 'short' in FFcyl.f must be set to .TRUE. and FFcyl.f recompiled. For a cavity with blades, the variable 'short' must be set to .FALSE. and recompiled. This will cause FFcyl to look for the scattering matrix files. Obviously if a shorted guide is desired, no scattering matrices are needed so jetsweep1 and jetsweep2 need not be run. To execute type:

```
FF<'filein'
```

where 'filein' is the input file shown below:

```
15      !Inlet Radius in cm
20      !Distance from mouth to handoff plane (cm)
1       !Include rim contribution (1:yes 0:no), for edge diffraction
-36.5,36.5,120 !Far Field Pattern: Theta(deg) start,end,number of steps
0,0,0     !Far Field Pattern: Phi(deg) start,end,number of steps
0,0,0     !Modulation from blade rotation: blade angle(deg)
!        start,end,number of steps
```

```
'file'      !File name header for scattering matrices and output files
2,18,150    !Frequency Sweep (GHz) start,end,number of steps
```

Note that 'distance from mouth to handoff plane' is the plane separating the blade region from the front of the cavity. This is the 'inlet distance from mouth to blades' in the 'file'.geo input file. For a shorted cavity, this distance is the cavity depth. Also modulation from blade rotation has not yet been tested and is left at 0,0,0. The output files of FF are designated 'file'.phi and 'file'.theta for phi and theta polarization, respectively. These files contain co-pol and cross-pol magnitude and phase information. By definition, the z-axis is the axis of propagation in the cavity, with theta measured from the z-axis. FFcyl assumes an inlet symmetric about z. In order to sweep the cavity symmetrically in aspect about the z-axis, simply set the theta start at a negative value. FFcyl will compute the same value for -theta as for +theta. Leave phi fixed at 0 degrees.

SAR images are created using the XPATCH Cifer post-processing routines [3]. To use these routines the output files of FF must be converted into an XPATCH 'file'.field format [3]. The mm2field.f fortran routine performs this conversion. mm2fld concatenates the 'file'.phi and 'file'.theta files over all frequencies from FFcyl into two 'file'.phi.data and 'file'.theta.data files and also converts the data into an XPATCH 'file'.field format for SAR processing using the XPATCH Cifer post-processing routines. This routine outputs an XPATCH .field file, 'file'.rng.field at a fixed aspect for conversion to an XPATCH 'file'.range for downrange viewing. To run type:

```
mm2fld<'file'.in
```

where 'file'.in format is:

```
121          !number of angles
2,18,150     !Frequency sweep(GHz): start,end,number of steps
sbncp1      !'file' of 'file'.phi and 'file'.theta
1           !Output of MM code(1) or do Phi,Theta Data Files Exist (2)
0,0         !Fixed Phi,Theta for Downrange Calculations
1           !Generate XPATCH field File? (1) YES;, (2) NO
```

Note that:

'Output of MM code (1) or do Phi,Theta Data Files Exist (2)' selection is 1 if input is 'file'.phi and 'file'.theta, 2 if input is concatenated 'file'.phi.data and 'file'.theta.data

'Fixed Phi,Theta for Downrange Calculations' will select fixed aspect for XPATCH .field file ('file'.rng.field) for conversion to an XPATCH .range file.

4 Examples

4.1 Engine Cavity/Straight Blades

This example is of the engine cavity with 8 straight blades mounted on a cylindrical flat hub. The hub and blades are 10 cm deep, mounted at the rear of the cavity, with the blades having an outer radius of 7.5 cm. The jetsweep1 and jetsweep2 input file sbncp1 follow along with the input file for The FF far field routines.

```
1      ! 1: straight 2: curved (blades)
2      ! plot interface fields (1:yes 2:no), Usually left at 2
2000   ! fstart (MHz), Frequency start
8950   ! fend (MHz), Frequency stop
50     ! delta f (MHz), Frequency step size
15.0   ! outer radius (cm.), Inner radius of cavity
7.5    ! inner radius (cm.), Outer radius of blades
20.0   ! inlet length from mouth to blades (cm)
10.0   ! blade length (cm)
8      ! number of blades
40.0   ! groove width - air (degrees), Angle between blades
2.5    ! first groove position (degrees), Measured from center of blade
0.25   ! far field angle step- theta (degrees)
1      ! rim diffraction 1:on 2:off
2      ! scattering matrix? (1:yes 2:no), Usually left at 2
30.0   ! total twist angle (degrees), For curved blades only
1D12   ! very large number (see code)
```

The groove position indicates the center of 1 blade is aligned with the x-axis. The groove width indicates there is a 40 degree gap between blades with each blade 5 degrees wide.

The input file for FF is as follows:

```
15      !Inlet Radius in cm
30      !Distance from mouth to handoff plane (cm)
1       !Include rim contribution (1:yes 0:no), for edge diffraction
-38.5,34.5,120 !Far Field Pattern: Theta(deg) start,end,number of steps
0,0,0    !Far Field Pattern: Phi(deg) start,end,number of steps
0,0,0    !Modulation from blade rotation: blade angle(deg)
!       start,end,number of steps
'barrelcm2a73' !File name header for scattering matrices and output
files
2,18,150 !Frequency Sweep (GHz) start,end,number of steps
```


Figure 2 shows the cavity configuration with Figures 2 and 3 showing RCS plots at several frequencies compared to measured data. Note that the data above 8.950 GHz was generated at single frequencies due to time constraints.

SAR images were also generated from the above input files. As indicated in the above input file, the images were generated from a data sweep of 2 to 8.950 GHz in 50 MHz steps. Figure 4 shows an image of the cavity with blades. The radar incidence is centered at -2 degrees theta with phi fixed at 0.0 degrees. Figure 5 shows the cavity with center radar incidence at -30 degrees theta and phi fixed at 0.0 degrees. Both images are VV polarization generated with the radar sweeping over a total aspect of 73 degrees centered as indicated previously for each image.

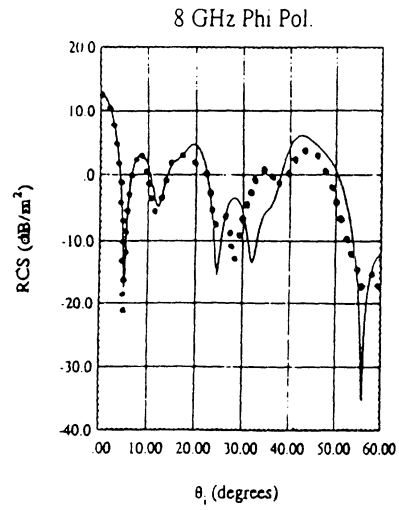
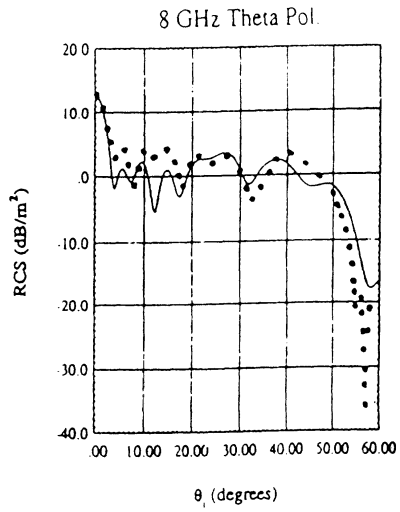
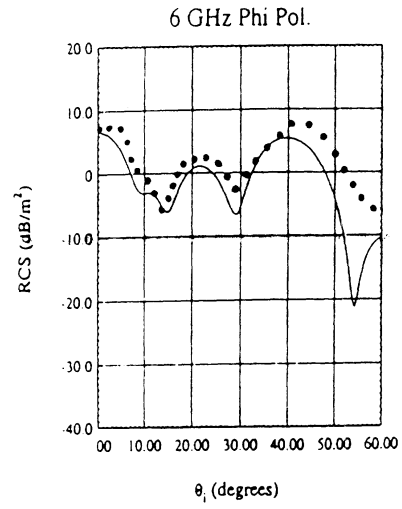
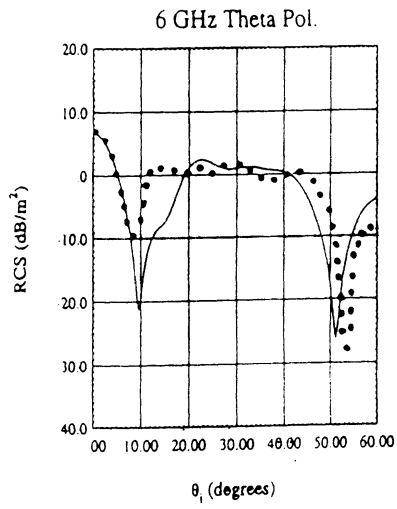
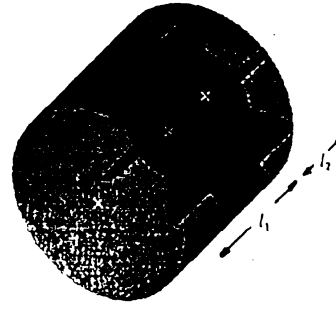
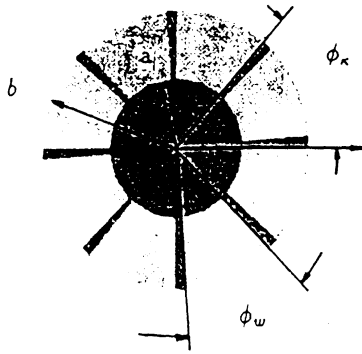


Figure 2: Engine with Straight Blades 1; RCS Plots

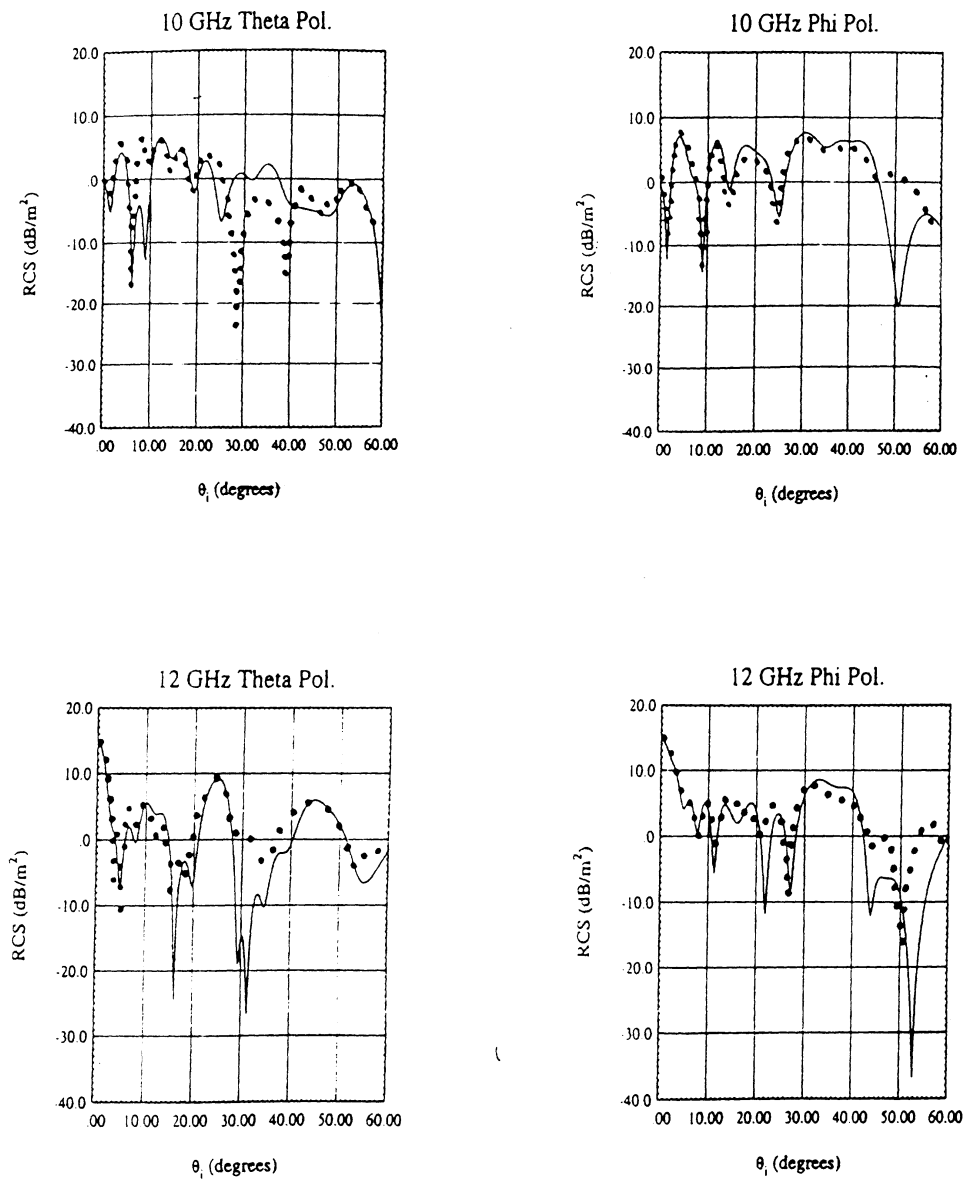


Figure 3: Engine with Straight Blades 2; RCS Plots

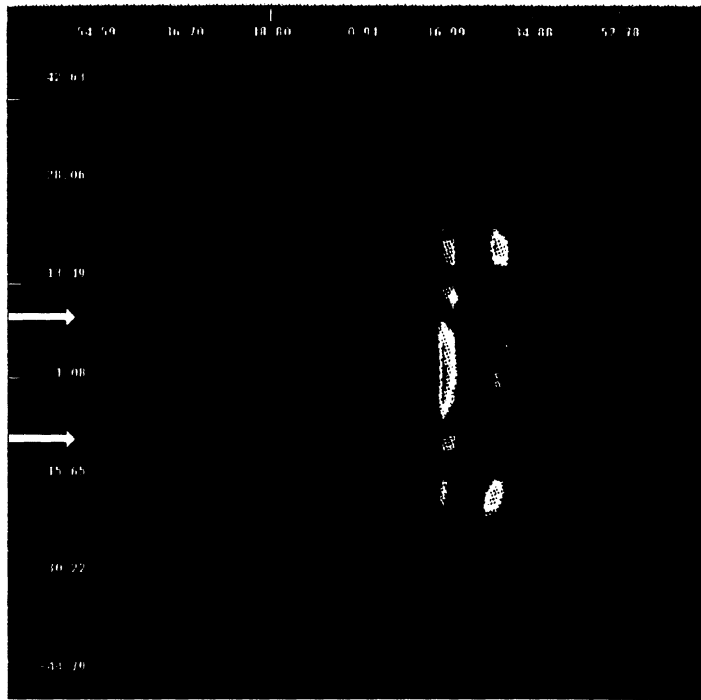


Figure 4: Sar Image-Engine with Straight Blades 2 Degrees off Axis

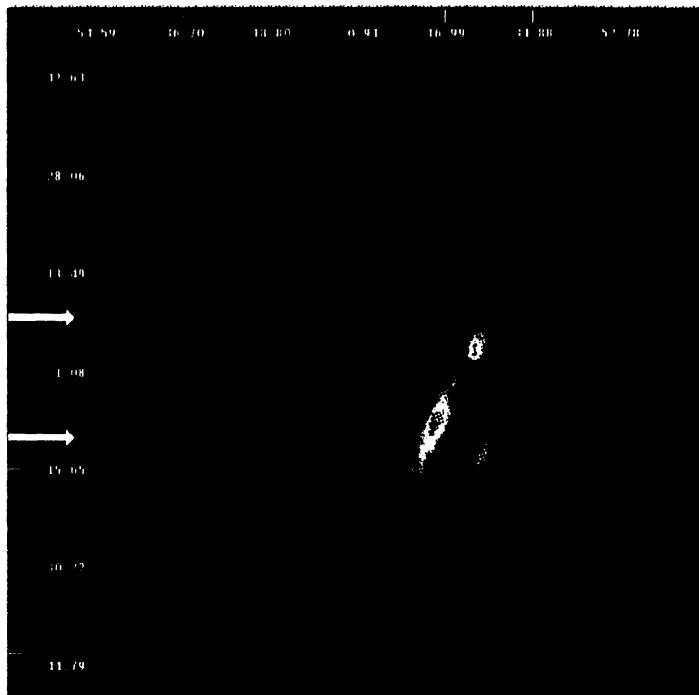


Figure 5: Sar Image-Engine with Straight Blades 30 Degrees off Axis

5 Installation

Installation is rather simple. In both, the jetsweep1 and jetsweep2 directories the file makget will compile the support routines necessary to run the code. Both directories contain the necessary files. Note that when running the scripts, jetsweep1 and jetsweep2, the main routines fsweep1.f and fsweep2.f are recompiled for each frequency and the support executables must be in the same directory as the script files. Simply type:

```
makget
```

in each directory to compile the supporting routines.

In the ffcyl directory again type makget to compile the supporting routines. In addition to compile FFcyl.f set the logical short to the appropriate state and type:

```
make -f Makefile.SGI FFcyl
```

Note that while sctmatcvt (converts 2D scattering matrices to 8D) is a script that recompiles the main matcvt.f for each frequency the FF manager for FFcyl is not and will be compiled with the makget make file. Following is a list of the jetsweep routines, separated by directory, with a brief description of the files contained therein.

All directories contain the following subroutines:

'file'.geo - Input file as described previously.

empfit.f - Determines max n/p traveling modes for given frequency.

getfhigh.f - These 3 routines feed initial frequency, high frequency,
getflow.f - and frequency step size to script files for automatic
getfdel.f - execution of routines.

param4.inc - Include file for main fortran routines. Dynamically updated
for each frequency by script files.

jetsweep1 directory:

The jetsweep1 routines compute modal inner products in parallel on the INTEL paragon and output an unformatted binary file in big endian format. These binary files are then stripped of their header for conversion to a Sun compatible binary format (little endian). The directory contains the following routines:

jetsweep1 - Script which automatically runs fsweep1.f routines through frequency. Gets frequency range from *.geo files, computes frequency dependent parameters for header file, then recompiles and runs main fsweep1.f routine. Only needs to be edited to define number of processors to use in parallel.

fsweep1.f - Main Fortran routine which computes modal inner products in parallel. Output is paragon binary 'file'_mat1_'fMHz'.

del4worker.f - Manager for parallelization.

del4_subs.f - Subroutines to compute Bessel functions.

jetcvtpara - Script which automatically runs fcvtpa.f through frequency. Similar operation to jetsweep1. Computes appropriate frequency dependent parameters, recompiles and runs fcvtpa.f.

fcvtpa.f - Fortran routine which strips header from paragon binaries for conversion to sun binaries. Also rewrites binaries in a format that can be read by sun routines which do conversion. Input is Output 'file'_mat1_'fMHz', output is 'file'_mat1_'fMHz'.sun

makget - Simple make file to compile empfit.f, getfhigh.f, getflow.f, getfdel.f, del4worker.f, and del4_subs.f

jetsweep2 directory

The jetsweep2 routines compute scattering matrices for use in the ffcyl far field calculations on a sun serial platform. The 'file'_mat1_'fMHz'.sun files from jetsweep1 are converted to sun binaries with fcvt_sun.f before running.

jetsweep2 - Similar to jetsweep1 for running the fsweep2.f routine.

fsweep2.f - Main Fortran routine which computes the scattering matrices. Output is ascii file 'file'_mat2_'fMHz' in 2D sparse matrix format.

del4_subs.f - Subroutines to compute bessel functions.

jetcvtsun - Similar to jetcvtpara.

fcvt_sun.f - Fortran routine which converts paragon binaries to sun compatible binaries. Input is 'file'_mat1_'fMHz'.sun, output is binary 'file'_mat1_'fMHz'.

byteswap.c - C routine which performs byteswap to convert big endian to little endian sun format. Called by fcvt_sun.f Courtesy of MSRC support people.

makget - Simple make file to compile empfit.f, getfhigh.f, getflow.f, getfdel.f, del4_subs.f and byteswap.c.

ffcyl directory

The ffcyl routines calculate far field scattering for a shorted termination or using the scattering matrices from jetsweep2 for a complex termination (turbine blades). Before running ffcyl for a complex termination the 2D scattering matrices generated by fsweep2.f must be converted to an 8D format by matcvt.f

FF.f - Fortran routine which generates input parameters for and runs FFcyl.f. When executed prompts user for input parameters.

FFcyl.f - Main fortran routine which computes far field scattering. Input for complex terminations is 'file'_'fMHz'.scatmat Output files for any termination are 'file'.phi, and 'file'.theta for phi and theta incident polarization respectively.

'file'.in - Input files for FF.f and mm2fld.f

bf3d.inc - Include file for FFcyl.f

slap_subs.f - Sparse Linear Algebra Package (SLAP), courtesy Mark K. Seager, Lawrence Livermore National Laboratory. Sparse matrix solver used by FFcyl.f

Makefile_SGI - Makefile for FFcyl.f To compile type 'make -f Makefile_SGI ff cyl'.

mm2fld.f - Routine to convert FFcyl output to XPATCH field format for SAR processing. Inputs are 'file'.phi, and 'file'.theta files and output is 'file'.field and 'file'.rngfld. 'file'.rngfld is single aspect field file for conversion to XPATCH 'file'.range. Also concatenates all 'file'.phi, and 'file'.theta files into 2 files, 'file'.phi.data and 'file'.theta.data. mm2fld can read in either separate or concatenated files.

concat - script called by mm2fld which sets up temporary parameter files and concatenates 'file'.phi, and 'file'.theta files.

sctmatcvt - Script which sweeps matcvt.f 2D-8D file format converter through frequency, recompiling for each frequency. Similar to jetsweep1 script.

matcvt.f - Fortran routine to convert 2D sparse scattering matrix to 8D. Input is 'file'_mat2.'fMHz' from jetsweep2, output is 'file'.'fMHz'.scatmat files for FFcyl input.

param5.inc - Include file for matcvt.f

makget - Simple make file to compile empfit.f, getflow.f, getfhigh.f and getfdel.f subroutines.

References

- [1] J. L. Volakis, H. T. Anastassiou and D. C. Ross, "Analysis of Jet Engine Inlets", Annual Report 030395-6-T, University of Michigan, Radiation Laboratory, December 1994.
- [2] J. L. Volakis, H. T. Anastassiou and D. C. Ross, "Parallelization of a Jet Engine Mode Matching Code", Annual Report 033279-1-T, University of Michigan, Radiation Laboratory, June 1995.
- [3] "User's Manual for xpatch", Vol. 1-4, DEMACO INC., June 1995.