100 Elwood Davis Road ♦ North Syracuse, NY 13212 ♦ USA

SonnetLab Pattern Data Exporter

Specialists in High-Frequency Electromagnetic Software
(315) 453-3096 Fax: (315) 451-1694 http://www.sonnetsoftware.com

Introduction

The pattern data framework that ships with the SonnetLab toolbox is designed to give Sonnet users the ability to export pattern data information from Sonnet.

The pattern exporter framework may be used as a standalone utility or used in conjunction with Sonnet's Matlab toolbox (SonnetLab). The interoperability layer between SonnetLab and the pattern exporter framework simplifies the process of extracting pattern data from projects already loaded in Matlab.

The pattern exporter framework includes a data reader which will load pattern data file information into Matlab structures. This simplifies the process of using Sonnet generated pattern data for Matlab calculations.

The SonnetLab pattern exporter can be used to do any of the following:

- Export pattern data from a Sonnet project file
- Export pattern data from a Matlab Sonnet project object
- Export pattern data from any set of analysis frequencies
- Specify the port excitations for any/all ports of the layout
- Plot pattern data to a Matlab figure

Requirements

The pattern export framework requires Sonnet 13 and SonnetLab version 5.0 or later. Before using the current exporter users will need to add the folder of scripts to their Matlab path.

General Instructions

Pattern data can be exported in two ways: the user may specify output arguments manually for each project that they wish to export data for or they may build one/more output configuration files that store common settings for use with any number of export operations.

There are two steps required in order to export pattern data from Sonnet projects. First the user must make a pattern data output configuration file and second they must execute a method that performs the request.

Pattern data output configuration files specify how pattern data should be outputted. Output configuration files specify options such as the theta and phi values, the excitation of ports, etc.

A single pattern data output configuration may be used for multiple iterations of a project design. For example if a user is optimizing a design and is generating a large number of similar project files they may be able to use a single output configuration for the entire set of project files.

The advantage of output configuration files is that a project's pattern data may be exported with a variety of options without the project being re-simulated.

Integration with SonnetLab

The pattern exporter can be called from SonnetLab with SonnetLab's built in exportPattern method. There are two ways to use the exportPattern() method: the first is to specify an output configuration file that specifies the pattern export options and the second way is to pass the appropriate values to the method such that it will build a temporary output configuration file. The syntax for calling the exportPattern() with a preexisting output configuration file is the following:

```
aData=Project.exportPattern(Filename);
```

The pattern data will be exported to a file with the same name as the project but with the extension ".pat" rather than ".son". The pattern data will be automatically read by the method and return an array of pattern data structures as described in the section titled "Read Pattern Output Data".

Instead of specifying an output configuration file users may instead specify values directly. When taking this approach the exportPattern() method will take the following arguments.

```
1) The PhiAngleVec [start stop step] of Phi (azimuthal angle)
   in degrees.
2) The ThetaAngleVec [start stop step] of Theta ("elevation"
   angle) in degrees.
3) The List of Frequencies at which the pattern should be
   calculated.
4) The port excitations/terminations.
   This should be a matrix with columns:
    [PortNumber Magnitude Phase(deg) Real(Z) Imag(Z)
     Inductance Capacitance]
   Example: [1 1 0 50 0 0 0]
   Which means: [Port 1, MAG=1, PHASE=0, R=50, X=0, L=0, C=0]
```

For example suppose we wanted to extract the pattern data at a frequency values of 2.3, 2.4, and 2.5, theta values from 0 to 360 in steps of 1, phi values from -90 to 90 in steps of 1, and an excitation of port 1 with MAG=1, PHASE=0, R=50, X=0, L=0, C=0.

```
>> aPatternData=Project.exportPattern([0 360 1],[-90 90 1],[2.3 2.4
2.5],[1 1 0 50 0 0 0])
```

When exporting pattern data using the SonnetLab method the user may specify the values for resistance, inductance, capacitance, reactance, and inductance for each port in the project as rows of a matrix where the columns are the port number, the excitation voltage, the phase, the real part of the resistance, the imaginary part of the resistance, the inductance and the capacitance.

```
[ PortNumber, Voltage, Phase Real Imag Inductance Capacitance;
  PortNumber, Voltage, Phase Real Imag Inductance Capacitance;
  PortNumber, Voltage, Phase Real Imag Inductance Capacitance;
  PortNumber, Voltage, Phase Real Imag Inductance Capacitance;
  PortNumber, Voltage, Phase Real Imag Inductance Capacitance; …]
```

The arguments for exportPattern() will effectively make a pattern request file, use it to export the current data and delete it. Using the exportPattern() method in the above manner may be simpler than creating a custom output configuration file for a single Sonnet project.

Export Data Using an Output Configuration File

The pattern data export tools may also be used standalone independent from the SonnetProject class. Existing output configuration files can be used to describe the output configuration of any number of Sonnet project files. To get the pattern data for a particular circuit the user simply needs to specify the export control file and the Sonnet project.

```
>> PatternExport(ControlFile,ProjectObjectOrFile);
```

The control file must be the name of an appropriate PG file on the hard drive. The project argument may either be the name of a Sonnet project file or a Sonnet project object existing in Matlab.

PatternExport() can optionally also accept a version number which indicates which version of Sonnet should be used to generate the pattern data file. The version of Sonnet should be at least thirteen.

```
>> PatternExport(RequestObjectOrFile,ProjectObjectOrFile,Version)
```

PatternExport() will temporarily modify the project such that Sonnet only simulates the project at the frequencies specified by the request object/file. The PatternExport() method will also temporarily enable current calculations for the project. The Sonnet project file will be reverted back to its original settings after the pattern data has been exported.

Read Pattern Output Data

When Sonnet exports pattern data it does so to a PAT file. The PAT file is a custom formatted document that stores export data. The pattern data exporter framework includes a reader which will import the pattern data into Matlab and store the data in an array of structures. The pattern data reader can be called to parse a PAT file with the following command:

```
>> aPatternData=PatternRead(theFilename);
```

Sonnet will export pattern data for each combination of frequency, theta and phi values. The pattern data reader will make a separate structure for every combination of said values; users can access the data they want by finding the structure in the array that contains data from the desired frequency, theta and phi values. A description of each of the fields in the output structure is presented in the table below:

Field Name	Description
Version	The user specified name for the export (Ex: ‘Original Design’)
Filename	The name of the PAT file that the pattern data was extracted from (Ex: ‘Project.pat)
ProjectName	The file name of the project (Ex: ‘Project.son’)
PatternControlBlock	Stores pattern export information including theta values, phi values, and port excitations.
PatternEntry	There is one pattern entry for each simulation frequency that pattern data was exported for. The pattern entries have a vector of the response for theta/phi value pairs.

Additional Framework Tools

The pattern exporter framework has several additional tools that demonstrate ways in which users can use the pattern exporter libraries and provide additional functionality.

Pattern Data Plot

The PatternPlot() method provides users with a function that will graphically plot a layout’s pattern data as a 3D Matlab figure. The below screenshot is the antenna pattern for the below patch antenna design.

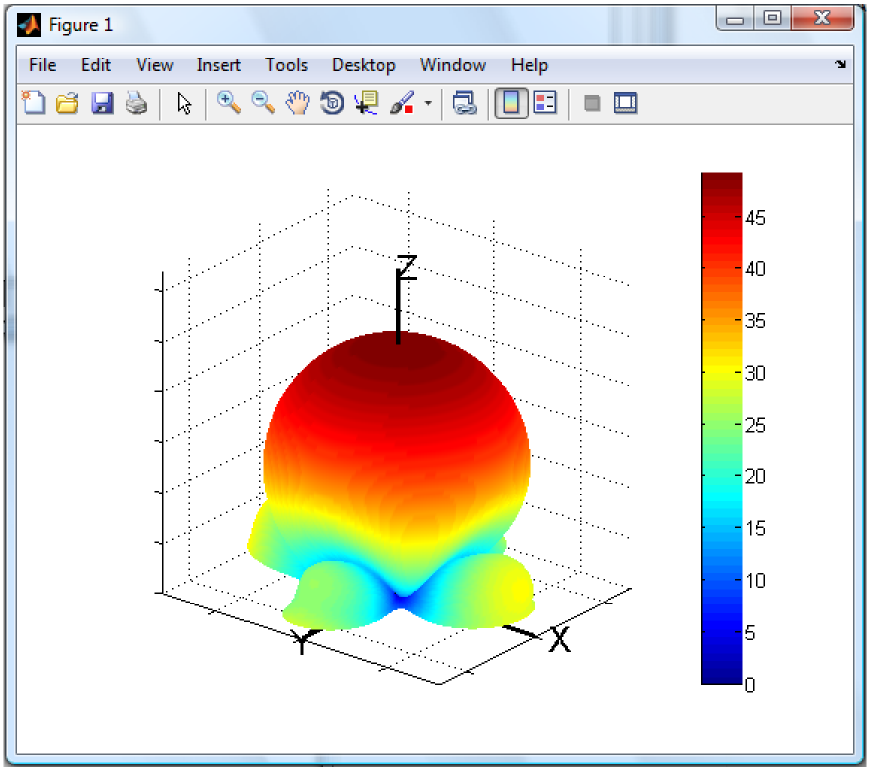
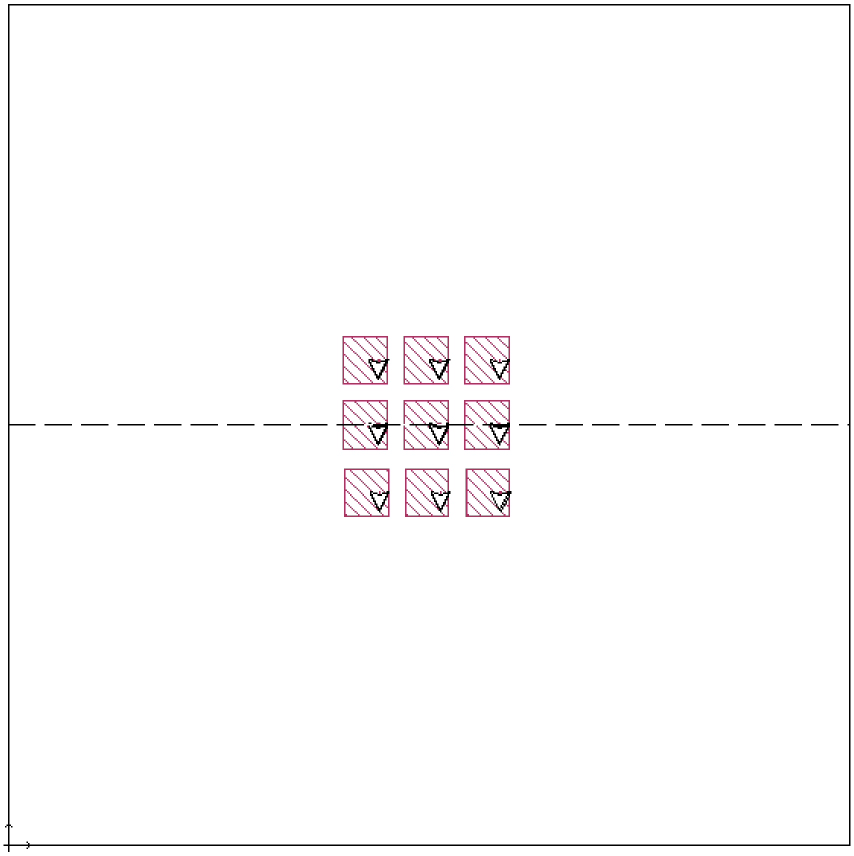


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Z
Y
X
45
40
35
30
25
20
15
10
5
0



Contact

Your feedback is important to us. If you have any questions or comments about SonnetLab, please contact Sonnet Support by email at support@sonnetsoftware.com.

Please make sure you are using the most up to date version of SonnetLab before submitting a bug report. When submitting a bug report please include the Sonnet project file that generated the error (Sonnet project files have the extension .son) and the output from the command "SonnetMatlabVersion". The more information that that we receive the faster it will be for us to resolve the issue and contact you back.