TNT 1.2 Users Guide

Bob Techentin
Mayo Special Purpose Processor Development Group

April 23, 2026

Table of Contents

List of Figures

Abstract

This users' guide describes the TNT transmission line modeling software package. The software was developed by the Special Purpose Processor Development Group, a research group at Mayo Clinic in Rochester Minnesota. This guide describes installation and operation of the TNT graphical user interface, and the MMTL and Wavelet simulators.

1 Introduction

The TNT software package was developed by the Special Purpose Processor Development Group (SPPDG) at Mayo Clinic in Rochester Minnesota. TNT was created in an effort to simplify and unify transmission line modeling for high performance electronics system designs. The SPPDG has developed and used several different transmission line "field solvers." Each tool had unique capabilities and limitations, and each tool had a different and often complex user interface.

TNT makes it very simple to create and modify two dimensional cross section descriptions of transmission line interconnect structures. The transmission line is represented as a cross section, with ground planes, dielectric layers, and conductors. Cross sections can be saved in a format called the Cross Section Description Language, which can easily be edited, customized, or even programmed.

TNT also makes it easy to run the electromagnetic field solvers which generate per-unit-length transmission characteristics. TNT is integrated with the Multilayer Multiconductor Transmission Line (MMTL) quasi-static simulator and two experimental wavelet-based full-wave transmission line analyzers. All simulators can be run simply from the TNT menus. MMTL can be run iteratively by sweeping cross section parameters or by iterating until a desired characteristic impedance is achieved.

2 License and Warranty

You are licensed to use, copy, modify, and share the TNT software package according to the terms of the GNU General Public License. The license terms are in the file COPYING included with TNT.

Under the General Public License, you may install and use TNT on any number of computers. You may examine the source code of the program to determine how it works. You may modify the program, hopefully making it better, and you may share the program with others. If you make improvements to TNT, we would appreciate it if you let us know.

Like most software, TNT comes with absolutely no warranty. Fortunatley, if there are problems, you are allowed to examine the source code, recompile the program, and correct the problem.

There are some software components included with TNT that are licensed separately. Printing on Microsoft Windows requires, for example, a print utility by Peter Lerup, which is licensed under terms described in the documentation in printfile215-32.zip. Tcl, Tk, and extensions are licensed under terms similar to those of the Berkely license, which imposes fewer obligations on the user than the GPL. Please refer to each package's license information.

3 Installation

TNT is free software. You can compile it from source code, or you can install a pre-compiled binary distribution. While most Windows users expect a binary distribution, many Unix or Linux users can build and install from sources. This section will describe both approaches.

TNT was originally developed to run on SPPDG workstations running HP-UX, and depends on several packages that are installed and maintained a the SPPDG. TNT has been ported to Linux and Windows, and an installation kit produced which should make it relatively straight forward to install the software on similar workstations or PCs.

3.1 Tcl/Tk and Other Dependencies

TNT requires a functional installation of Tcl and Tk with several extensions, including BWidget, Incr Tcl, and Iwidgets. You will need to install all of these packages in order for TNT to function correctly. TNT expects to be able to find the `wish` windowing shell in your command path.

You *could* compile and install these packages from the freely available source code hosted at Sourceforge. Specifically, you can acquire them from http://tcl.sf.net/, http://tcllib.sf.net/, and http://incrtcl.sf.net/. You will need an ANSI C compiler, such as the GNU Compiler Collection from http://gcc.gnu.org. You should be able to use the Microsoft Visual C++ compiler or MinGW for Windows systems. You may find this an enjoyable challenge, or, if you are not a programmer, you might find the process difficult and frustrating.

Alternatively (and highly recommended), you can install a freely available, pre-compiled Tcl and Tk package from ActiveState, which includes all these extensions. The ActiveTcl distribution is, in fact, installed on SPPDG workstations and PCs. Unfortunately, while the ActiveState packages are free to use, we are not permitted to distribute the packages with TNT. You can obtain ActiveTcl for Windows, Linux, Solaris, or HP-UX from http://www.activestate.com/.

3.2 Microsoft Windows Installation

3.2.1 Windows and Tcl/Tk

The Windows distribution includes a Tcl/Tk runtime along with all the extensions necessary to run the application. So you don't really need to download and install ActiveTcl or other Tcl/Tk distributions to make TNT work on your PC. But Tcl is so much fun to use, that you really ought to consider getting a copy.

3.2.2 Windows and TNT

TNT has a basic Windows installation program. Just click on the installer file, accept the GPL license, specify an installation location, and let it do the work. Installing TNT does not require administrative priviledges on Windows 2000 or Windows XP, but if an administrator performs the installation, then the package will be available for all users.

The installation program will create a shortcut to run TNT. You may want to copy that shortcut, and modify the "Start In" directory to a location where you normally work with transmission line simulations.

3.3 UNIX Installation

3.3.1 UNIX and Tcl/Tk

Download and install ActiveTcl from http://www.activestate.com/. This gets you Tcl, Tk, BWidget, Incr Tcl, Iwidgets, and a lot of other cool stuff.

3.3.2 UNIX and TNT

If you have a binary distribution, just copy the TNT directory tree from the installation kit to a location from which applications are normally run. This could be `/usr/local/` or a personal subdirectory or some other location. Make sure that the directories for `wish` and `.../tnt/bin` are in your path.

To build TNT from sources, unpack the source code distribution archive, and follow the instructions in the file named INSTALL.

4 Starting TNT

On Windows, the TNT program is run by clicking on the menu item or shortcut that you created in 3.2.2. Alternatively, you can use Windows Explorer to navigate directly to the TNT installation directory, and click on `tnt.tcl`.

On Unix, the program is invoked at the command line by executing either "tnt" or "tnt.tcl" from the installation directory. If the binary directory `.../tnt/bin` is included in your path, simply type the command. Otherwise, you may specify a full path name to `tnt.tcl`, or you can change your current working directory (cd) to the `.../tnt/bin` directory, and run the application from there.

When started, TNT will not have any simulation parameters defined. You may use the **Open**, **Save**, and **Save As...** options from the **File** menu to load and save TNT cross section files.

4.1 TNT Main Window

The TNT main window contains an application menu bar, buttons for creating new cross section structures, a layer stackup, and a drawing of the cross section.

The menu bar has several pull-down menus. The **File** menu has options for opening, saving, and printing cross section description files. The **View** menu toggles display options. Materials lists can be re-loaded using options on the **Setup** menu. MMTL simulations can be run from the **BEM**, **Sweep**, and

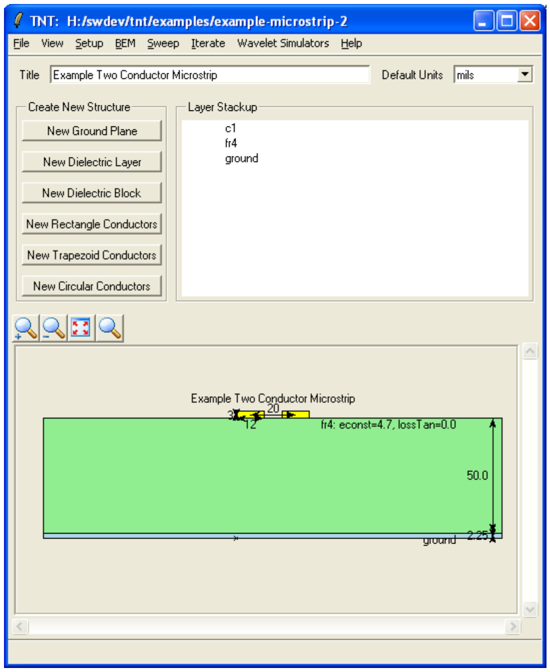

Figure 1: TNT Main Window

Iterate menus. The full-wave experimental wavelet based simulators can be run from the **Wavelet Simulators** menu.

5 Cross Sections

5.1 Example Cross Sections

You can create a cross section from scratch, as described in section 5.2, or you can start with one of the example cross sections that is distributed with TNT, and modify it according to your needs.

The example cross sections are in the TNT installation directory, under `.../TNT/examples`. Choose **File** ⇒ **Open** from the menus, navigate to the example directory, and choose one of the cross section files. You may then modify it and save it elsewhere with **File** ⇒ **Save As**.

5.2 Creating a Cross Section

To create a new cross section, you can choose **File** ⇒ **New** from the menu. This will clear any existing structures from the layer stackup and the drawing window.

Enter a model title in the **Title** field. The model title should be descriptive, and will be saved with the cross section model. Set the units to match the physical dimensions of your transmission line structure.

Start adding structures to the layer stackup by creating a ground plane, then adding layers of dielectrics and conductors. Clicking on the **New Ground Plane** button will open a new window allowing you to define the structure name and the thickness. Clicking **Add** on the dialog will add the ground plane to the layer stackup and the drawing.

Continue by adding dielectric layers. Click on the **New Dielectric Layer** button to open the dielectric properties entry form. Each layer has a name, thickness and material characteristics. Make sure that the default units selected at the top of the screen match your intentions for the layer dimensions. (You don't really want a 42 meter thick dielectric, do you?)

Layers will appear in the cross section drawing as you add them, and layer names will appear in the layer stackup. If you do not add a top ground plane (i.e., microstrip), then air is assumed to be above the top defined layer.

Add conductors to the cross section by selecting one of **New Rectangle**, **Trapezoid**, or **Circular Conductors** buttons, which will open a new properties dialog for the conductors. Each conductor structure has dimensions and material properties. When you **Add** these conductors, they will rest atop the last defined dielectric layer. You can add a single conductor, or a group of identical conductors with a specified pitch. You can specify X and Y offsets to the conductors.

You can also define **New Dielectric Block**s which are arbitrary rectangles of dielectric material. These blocks can be used to define non-planar (conformal) dielectric structures.

5.3 Modifying a Cross Section

Select cross section elements by clicking on them either in the **Layer Stackup** window or on the cross section drawing. Double-clicking on a cross section element opens its properties dialog. You can choose to **Modify** the properties or **Delete** the structure.

You can rearrange the order of the layer stackup by clicking and dragging structures in the **Layer Stackup** window.

The cross section drawing is supposed to give you a graphical depiction of your transmission line cross section. Unfortunately, the scale of many conductor definitions makes it difficult to see the drawing easily. You may need to use the zoom buttons to get a better view of the structures.

5.4 Printing

You can print the cross section picture on a postscript printer by choosing **File** ⇒ **Print** from the menu. The print dialog offers several options, including paper size, orientation, and output specification.

On Unix systems, the default printer command is **lpr**, which should work for most installations. Choose a different print command, or add options, if you choose. On Windows systems, the default printer command is **PrFile32.exe**, which is a small utility program that directs the postscript to a Windows print queue. If you do not have a postscript printer, you will likely get many pages of printed postscript commands.

6 BEM MMTL

The Boundary Element Method (BEM) Multilayer Multiconductor Transmission Line (MMTL) simulator is the most recent of several generations of electromagnetic modeling packages developed at the SPPDG. This program uses the Method of Moments (MOM) technique to quickly compute capacitance and inductance parameters for a transmission line structure.

BEM MMTL is loss free, and makes use of the so-called "Quasi-TEM" assumptions. It is assumed that the electromagnetic fields are substantially transverse to the direction of propagation "down" a transmission line. For typical printed circuit board (PCB) geometries, BEM MMTL should be accurate up to about 5 GHz. This frequency limit scales with geometry and materials, so BEM MMTL should give good results at higher frequencies for smaller geometries and lower losses. Pure Transverse Electromagnetic (TEM) propagation does not exist, since all materials contains some loss. However, in most PCB, multichip module (MCM) and integrated circuit (IC) designs, losses are relatively small, and the "Quasi-TEM" assumptions apply.

MMTL requires a bottom ground plane. A top ground plane (for stripline) is optional. The ground plane name is not particularly important, and thickness is irrelevant. This is a perfect ground plane.

Dielectric permittivity (relative dielectric constant) is used to compute capacitance and inductance. BEM MMTL does not use the dielectric loss tangent or the conductor conductivity.

If a conductor name starts with "G" or "g", it is a special "ground" conductor. This conductor will be considered as absolute ground, and will not appear in the MMTL results. This feature is very useful for defining coplanar waveguides.

6.1 Running BEM MMTL

Choose **BEM** ⇒ **Run BEM MMTL Simulation** from the TNT menu to run the simulator. You will be prompted to enter several values to control the simulation.

Coupling Length (in default units) and **Risetime** are used to compute crosstalk estimates for the transmission line segment. The matrix of MMTL results is presented in per-meter values, and the circuit parameters are considered frequency independent.

Conductor and **Dielectric Segments** are mesh parameters that BEM MMTL will use to discretize the cross section components. Conductors and dielectric segments can be meshed with different resolutions. These values control (to some extent) the simulation accuracy. Larger numbers may give a more accurate answer, but also result in more computation time. Really large numbers (like 100) are not recommended.

Clicking on **Run** will save the cross section file and run the simulator. A log window will be displayed that will show the simulator output. MMTL will list values that it reads from the cross section file, and print messages as it performs various computations.

The output will include "Asymmetry Ratios" for both the inductance and electrostatic induction matrices. The MoM algorithm employed by MMTL does not have a precise error computation, but it does check the output parameter matrices for symmetry. If the values are asymmetric, it may indicate that more (or sometimes fewer) conductor or dielectric segments should be specified for the simulator.

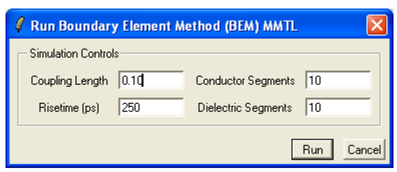

Figure 2: Running BEM MMTL

6.2 Exporting HSPICE W-Element

Choosing **BEM ⇒ Generate HSPICE W** from the menu will generate a W-element model from the MMTL results. The model file will have the same name as your cross section file, with the extension `.hspice-w.rlgc`. TNT will open a new window to show you the generated W-element model. You can cut and paste from this window, or refer to the generated file from other applications.

The W-element file contains a comment header that describes the original cross section and MMTL run. Matrices L_0 and C_0 are copied directly from MMTL's inductance and electrostatic induction matrices.

Conductor losses, R_0 and R_S are estimated from the conductor geometries and materials properties, with these formulations.

$$R_0 = R_{DC} = \frac{1}{\sigma C_{\text{Area}}} \ \Omega/m \tag{1}$$

$$R_S = R_{AC} = \frac{\sqrt{\pi \mu_0/\sigma}}{C_{\text{Circum}}} \ H/m \tag{2}$$

Where σ is the conductivity, C_{Area} is the conductor area, C_{Circum} is the conductor circumference, and μ_0 is the permeability of free space.

6.3 Parameter Sweep

TNT can run BEM MMTL iteratively, sweeping one or more parameters through a range of values. Choosing **Sweep** ⇒ **Sweep Simulation** from the menu will allow you to choose the parameters to be swept, from a dialog similar to that shown in Figure 3. Note that you can select any number of parameters, including the simulation control parameters normally found on the BEM MMTL simulation control dialog shown in Figure 2.

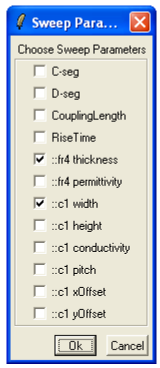

Figure 3: Choosing BEM MMTL parameters to sweep

Once you have selected parameters to sweep, you must specify starting and ending values and the number of iterations for each parameter. A dialog similar to Figure 4 will allow you to enter the controlling values.

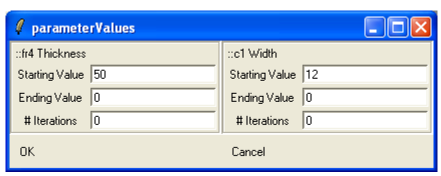

Figure 4: Specifying BEM MMTL Sweep Ranges

Sweeping several parameters can result in a very large number of simulations. TNT will run a comprehensive sweep, including N runs, where N is the product of the number of iterations of each selected parameter. If you choose ten iterations of each of three different parameters, you should expect 1000 MMTL runs. TNT prompts you one last time with the total number of iterations, to give you one last chance to bail out.

Once the simulations are run, you can view the results of all the simulations, or write a "character separated file" (sometimes called a "comma separated file" or csv), which is suitable for import into a spreadsheet program for analysis. All parameters are exported to the csv file.

6.4 Iterating Conductor Width

Iteration is a specialized form of parameter sweep. For some transmission line designs, all layer thicknesses and materials properties are fixed, and the engineer has control only over line width. The iteration feature of TNT allows you to specify these basic cross section parameters, and then run MMTL iteratively until a certain characteristic impedance is obtained.

7 Wavelet Simulators

TNT includes two prototype wavelet-based transmission line simulation tools. These tools use a simple finite element approach with Coifman wavelet basis functions to compute full-wave transmission line parameters. You may choose either the "RL" calculator for resistance and inductance, or the "CAP" calculator for capacitance.

The "RL" calculator dialog allows you to enter a number of frequency points at which you would like the parameters computed. Simply enter the frequencies, in hertz, separated by spaces (e.g., "1e9 2e9 3e9"). The "CAP" calculator simply computes the line-to-line capacitance, which is relatively independent of frequency.

These simulators have not been completely generalized. They are limited to a single ground plane, so stripline simulations will be incorrect. They also are also limited to planar dielectric layers, and will crash if attempting to simulate rectangle dielectrics.

8 On-Line Help

TNT includes online help in the form of this users guide and other documentation. The online help is displayed in a web browser. On Unix systems, TNT will attempt to run firefox, opera, mozilla, and netscape, in that order, to display the files. On Windows, TNT will attempt to launch the default web browser on the document.

Printable versions of these documents are also available in TNT's `doc` directory, in portable document format (PDF).

9 Acknowledgments

The authors would like to thank the engineers of Mayo SPPDG for their patience, feedback, and technical support; Dr. George Pan of Arizona State University and the students of his signal propagation laboratory for algorithm research and prototype code; N. Naclario, Z. Lemnios, D. Healy, D. Cochran, A. Krishnan, DARPA, and C. Hanson, SPAWAR/Code 8505, for program support.

References

[1] Pan, G-W, K. S. Olson, and B. K. Gilbert, *Improved Algorithmic Methods For the Prediction of Wavefront Propagation Behavior in Multiconductor Transmission Lines for High Frequency Digital Signal Processors.* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 8(6)608-621 (June) 1989.

[2] Pan, G. W., J. A. Prentice, S. K. Zahn, A. J. Staniszewski, W. L. Walters, and B. K. Gilbert, *The Simulation of High-Speed, High-Density Digital Interconnects in Single Chip Packages and Multichip Modules,* IEEE Transactions on Components, Hybrids, and Manufacturing Technology, 15(4):465-477 (August) 1992.