# Two-Dimensional TM and TE FDTD Codes with Visualization

Allen W. Glisson[1]
aglisson@olemiss.edu

Atef Z. Elsherbeni[1]
atef@olemiss.edu

Chun-Wen P. Huang[2]
cwhuang@anadigics.com

[1]Dept. of Electrical Engineering
University of Mississippi
University, MS 38677

[2]Anadigics, Inc.
35 Technology Drive
Warren, NJ 07059

## Introduction

This paper describes the use of two available two-dimensional Finite Difference Time Domain (FDTD) codes with visualization capabilities that have been developed primarily for educational use. The programs animate time domain scattering by a two-dimensional material geometry excited either by a *z*-directed electric current line source in the TM case (*tmpml*) or by a *z*-directed magnetic current line source in the TE case (*tepml*). A fairly general scattering geometry can be described by defining various regions with different isotropic constitutive parameters. The perfectly matched layer (PML) absorbing boundary condition is applied at the computational boundaries. The codes are available from the ACES web site http:/aces.olemiss.edu.

## Code Components and System Requirements

The two programs and their associated example input data files comprise the following ten files:

| | |
|---|---|
| tmpml.exe | tepml.exe |
| tmpml.dim | tepml.dim |
| tmpml.geo | tepml.geo |
| tmpml.vwc | tepml.vwc |
| tmpml.ind | tepml.ind |

The codes were compiled with Compaq Visual Fortran version 6.6 and use the Compaq Array Viewer version 1.6. They have been tested under Windows 2000. In addition, if the machine on which the programs are to run does not have the Compaq Array Viewer installed, the user must obtain and install the Compaq Array Viewer Demo program. This "demo" is available on the web at http://www.compaq.com/fortran/

If the user already has the Compaq Array Viewer installed, he should verify that the DLL file aview160.dll is in the Windows System directory.

## Program Usage

The programs *tmpml.exe* and *tepml.exe* are “Fortran Console Applications” that run in a Command Prompt window. The user interacts with the FDTD program through keyboard input to the Command Prompt window. However, when so instructed by appropriate input data, the program invokes the Compaq Array Viewer for visualization of the solution. The Array Viewer program is a native Windows program. The user can interact with the Array Viewer program as with most Windows programs to change viewing angle, display method, plot parameters, etc.

The programs *tmpml.exe* and *tepml.exe* can be run by any of the usual methods for executing Windows programs.

## Input Data Description

The descriptions of the input data files below are given explicitly for the TM program *tmpml.exe*. The data files for the TE program *tepml.exe* are identical except that they begin with the letters “te” rather than “tm” and except for a few minor differences in the data definitions, which are identified in the descriptions when appropriate.

**Filename**: *tmpml.dim* — dynamic dimensioning parameters

Data required for dynamic dimensioning of the program arrays must be present in this data file in the form shown in Figure 1 (free format):

```
      Ngx           Ngy
      NPMLlayers    PMLpower
      Nax           Nay
```

**Figure 1. Data format for the file *tmpml.dim.***

Ngx and Ngy represent the numbers of “grid points” in the *x* and *y* dimensions of the computational grid in the TM case. These grid points are defined to be the points at which the *z* component of the electric field is computed. In the TE case, `Ngx` and `Ngy` represent the numbers of rectangular “cells” in the *x* and *y* dimensions of the computational grid. The center points of these rectangular cells are the points at which the *z* component of the magnetic field is computed in the TE case. All non-free-space materials of the scattering geometry reside within this computational grid. There is always one layer of free-space material surrounding the computational grid in addition to the PML region. `Nax` and `Nay` are the numbers of additional free-space layers to include between the computational grid and the PML region. `NPMLlayers` is the number of PML layers to be used to absorb the outgoing waves. `PMLpower` is the exponent in the expression for the rate of increase of $\sigma$ as a function of distance from the air-PML interface within the PML region, typically 1, 2, 3, or 4. `PMLpower` is a real number; all other quantities in this file are integers.

**Filename**: *tmpml.vwc* — Array Viewer control parameters

Data required for initial control of the Array Viewer program must be present in this data file in the form shown in Figure 2 (free format):

```
IGeomDraw
IFieldDraw
IPlotStep
IPlotPause
PlotPeak
IFldComp
```

**Figure 2. Data format for the file *tmpml.vwc.***

`IGeomDraw` controls whether or not the Array Viewer is used to display a schematic representation of the geometry before computation begins. A value of 1 turns geometry visualization on, and a value of 0 turns it off. `IFieldDraw` controls whether or not the Array Viewer is used to display the value of a field component during execution. A value of 1 turns field visualization on, and a value of 0 turns it off. If field visualization is on, the remaining values will control the initial visualization parameters.

`IPlotStep` controls the plot update frequency. If `IPlotStep` is set to a value of *n*, the field plot will be updated every *n*th time step. The more often the field plot is updated, the slower the simulation will run. `IPlotPause` controls the number of time steps before execution is paused. If `IPlotPause` is set to a value of *n*, the simulation will be paused at every *n*th time step. `IPlotPause` is useful to allow the user to interact with the FDTD program (instead of the Array Viewer program). When the FDTD program is paused in this manner, the user can take time to view the field plot more carefully, or can interact with the FDTD program to change the current values of `IPlotStep` and/or `IPlotPause`. To interact with the FDTD program, or even to continue execution from the paused state, however, the Command Prompt window must be the “active window.”

The variable `PlotPeak` is used to set the maximum magnitudes of the field values to be displayed by the Array Viewer program.

The variable `IFldComp` is used to choose which field component is to be displayed. For the TM case the choices are 1 for $E_z$, 2 for $H_x$, or 3 for $H_y$. The peak value variable `PlotPeak` is then used to determine the peak value of $E_z$ that would be displayed. If $H_x$ or $H_y$ is chosen for display instead, the value of `PlotPeak` is automatically adjusted by the free-space impedance value $\eta_0$. For the TE case the values of `IFldComp` for the various component choices are 1 for $E_x$, 2 for $E_y$, or 3 for $H_z$. The peak value variable `PlotPeak` in this case is used to determine the peak value of $H_z$ that would be displayed. If $E_x$ or $E_y$ is chosen for display instead, the value of `PlotPeak` is automatically adjusted by $\eta_0$. `PlotPeak` is a real variable; all other variables in this file are integers.

**Filename**: *tmpml.geo* — scatterer geometry data

Data required to describe the scatterer geometry must be present in this data file in the form shown in Figure 3 (free format):

```
NumMatDefs
MatID        epsr      mur     sigmae    sigmam
MatID        epsr      mur     sigmae    sigmam
      :
      :
      :
MatID        epsr      mur     sigmae    sigmam
ID           nxstart   nxend   nystart   nyend
ID           nxstart   nxend   nystart   nyend
      :
      :
      :
ID           nxstart   nxend   nystart   nyend
-1           0         0       0         0
```

**Figure 3. Data format for the file *tmpml.geo.***

The first line of this data file identifies the number of different materials (NumMatDefs) to be defined to the program in subsequent lines of the data file. NumMatDefs must be between 1 and 50, allowing for use of 50 different materials as part of the scatterer geometry. Following the first line there must be NumMatDefs lines of data describing the materials. These subsequent data lines must specify the integer material identification number (MatID), the relative permittivity $\varepsilon_r$ (epsr), the relative permeability $\mu_r$ (mur), the electric conductivity $\sigma^e$ (sigmae), and the magnetic conductivity $\sigma^m$ (sigmam). Thus, the first NumMatDefs+1 lines of this data file represent material definition data available to the program. Note that these are just material definitions; it is not necessary for each material defined here to be used in the scatterer geometry to be simulated.

In addition to the definitions noted in the preceding paragraph, the free-space material type is predefined with a material ID of 0, while the perfect electric conductor (PEC) material type is predefined with a material ID of 51. These two material types can be used in the definition of the scatterer along with the other material types provided by the user. Any or all of the defined material types can be used in a particular scatterer geometry.

The user must define the scatterer geometry starting in line NumMatDefs+2 of this data file. The geometry is defined in rectangular blocks by specifying a material ID (ID) and the range occupied by the material in the *x* and *y* dimensions in terms of the computational grid indices nxstart, nxend, nystart, and nyend. For the TM case, note that these quantities represent *node* or *grid point* values where the *z* component of the electric field is computed. Thus, in the TM case, the materials are specified to fill the spatial region with the corners defined by the four specified nodes. For the TE case, the quantities nxstart, nxend, nystart, and nyend represent *cell* number values where the *z* component of the magnetic field is computed. Thus, in the TE case, the materials are specified to fill the entire spatial region with the corners defined by the outermost corners of the four specified cells.

Each successive geometry specification line overwrites any material type previously specified for the indicated region. Thus, for example, one can create a hollow, rectangular dielectric cylinder by first creating a solid dielectric cylinder with the exterior dimensions desired, and then specifying the interior hollow region to have material ID 0. The interior region that was originally specified as dielectric material will be replaced with the new material type of free space.

Scatterer geometry input is terminated by specifying a material type of –1 (negative one; the computation grid ranges must still be present as suggested by the data file form shown in Figure 3, but they are not used). When the material type of –1 is encountered, no further data lines are read by the program.

**Filename**: *tmpml.ind* — other input data

Other input data required to describe the number of time steps to use, the spatial increments, the excitation, the saved field values, the output data file names, etc., must be present in this data file in the form shown in Figure 4 (free format):

The descriptions of the input variables in this file follow:

`Nstop` Final time step number to use.

`JzAmp` For the TM case this value represents the excitation electric current line source (Real, signed) amplitude. For the TE case this is instead `MzAmp` – the excitation magnetic current line source amplitude.

```
Nstop     JzAmp
dx          dy
Nsourcex    Nsourcey
ExciteType
MDecayFactor
SineFreq
PMLsxMax
PMLsyMax
DiagFilename
NoutFiles                                 _________
Outfilename                                        |   This section is repeated
Header                                             |      NoutFiles times
NTestPts                                           |   (once for each output file)
NPx         NPy        NfldType                    |
NPx         NPy        NfldType                    |
      :                                            |
      :                                            |
      :                                            |
NPx         NPy        NfldType   _________________|
```

**Figure 4. Data format for the file *tmpml.ind.***

`dx,dy` Spatial increment values (Real) in $x$ and $y$ in meters.

`Nsourcex,`
`Nsourcey` For the TM case these values represent the *node* indices in the $x$ and $y$ directions within the computational grid of the electric current line source $J_z$. For the TE case they represent the *cell* indices in the $x$ and $y$ directions

within the computational grid of the magnetic current line source $M_z$. The magnetic current line source is located at the center of the specified cell.

`ExciteType` Excitation type (Character): there are two allowed current source excitation types that must be entered starting in column 1 *exactly* as shown in one of the following forms:

| | |
|---|---|
| `sine` | A sine wave starting at $t$=0 |
| `Gaussian` | A Gaussian pulse |

`MDecayFactor` Integer $M$ that controls the decay rate of the Gaussian pulse. It is used to compute the inverse Gaussian decay constant $\tau = M\,\max(\Delta x, \Delta y)/(2c\sqrt{3})$. A value must appear in the data file for this variable in all cases, but is used only for the "Gaussian" excitation. $M$ should be chosen such that $M \simeq \sqrt{3}\,n_c$, where $n_c$ is the number of cells to be used per wavelength at the maximum usable frequency.

`SineFreq` Sine wave frequency in GHz. A value must appear in the data file for this variable in all cases, but is used only for the "sine" excitation.

`PMLsxMax`,
`PMLsyMax` The maximum values of the electric conductivity to be used in the PML region in the $x$ and $y$ directions.

`DiagFilename` The name (character*16) of the diagnostic output file.

`NoutFiles` Number of output data files (maximum of 10).

`Outfilename` The name (character*16) of the data output file.

`Header` The header string (character*70) to appear in the data output files.

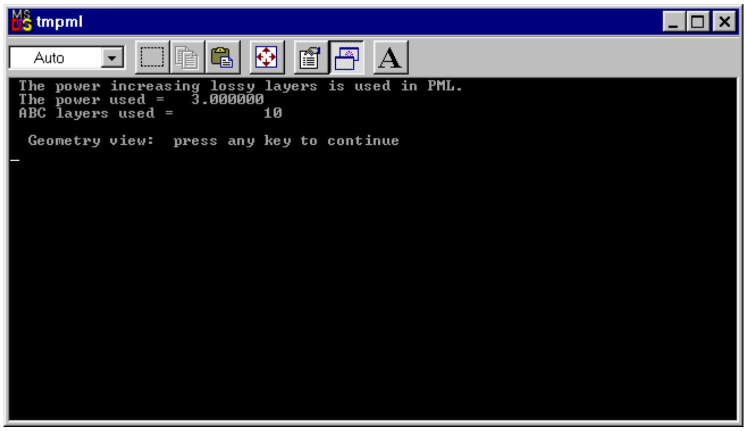
`NTestPts` The number of test points at which to save and print the field values (maximum of 500).

`NPx`, `NPy`,
`NFldType` For the TM case `NPx` and `NPy` specify the *node* indices in the $x$ and $y$ directions within the computational grid at which the field value is to be saved for printing. `NFldType` specifies which field component to save: 1 for $E_z$, 2 for $H_x$, or 3 for $H_y$. For $H_x$ and $H_y$, which are located between "nodes," the value saved is the next value encountered in the

increasing index direction from the node (`NPx, NPy`). For $H_y$, for example, this will be the value of $H_y$ at the point represented by (`NPx+1/2, NPy`). For the TE case `NPx` and `NPy` specify the *cell* indices in the *x* and *y* directions within the computational grid at which the field value is to be saved for printing. `NFldType` specifies which field component to save: 1 for $E_x$, 2 for $E_y$, or 3 for $H_z$. For $E_x$ and $E_y$, which are located between "cells," the value saved is the next value encountered in the increasing index direction from the center of cell (`NPx, NPy`). For $E_y$, for example, this will be the value of $E_y$ at the point represented by (`NPx+1/2, NPy`).

## Code Execution

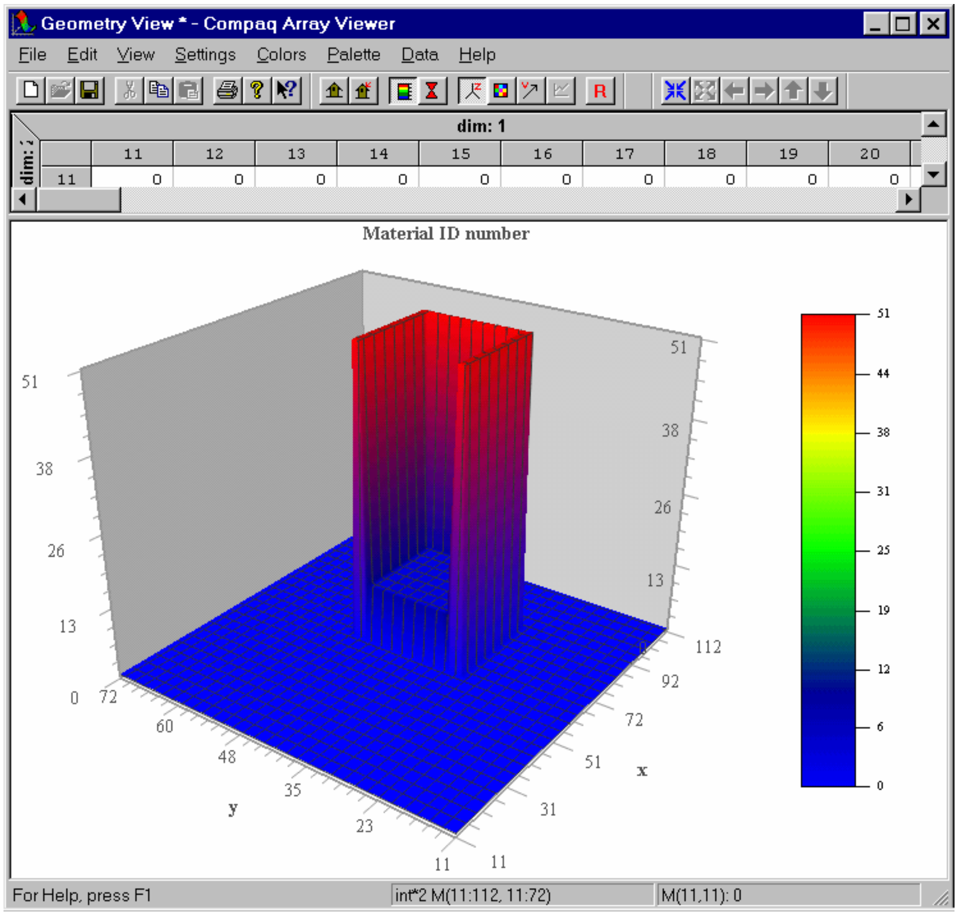
When the code begins execution, a Command Prompt window similar to that shown in Figure 5 will appear. If `IGeomDraw` has been set to a value of one in the data file *tmpml.vwc* (*tepml.vwc* for the TE case), the geometry view window will also appear (Figure 6) and will be the active window. In the geometry window in Figure 6 the material ID number is plotted as a function of *x* and *y*. The sample geometry shows a 3-sided PEC box (material ID 51) that is filled with a material medium (material ID 10), and the remainder of the space is filled with homogeneous free space (material ID 0).



```
The power increasing lossy layers is used in PML.
The power used =    3.000000
ABC layers used =            10

 Geometry view:  press any key to continue
_
```

**Figure 5. The *tmpml* Command Prompt window.**

With the geometry window open, the user may rotate the schematic geometry view interactively with the mouse to obtain different views. The user may also interact with the menu items and tool bars of the Array Viewer program to zoom in to view a smaller region of space (i.e., the material ID array), change the plot scale, change to an image map view of the array as shown in Figure 7, or change other Array Viewer options.



| dim: 2 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

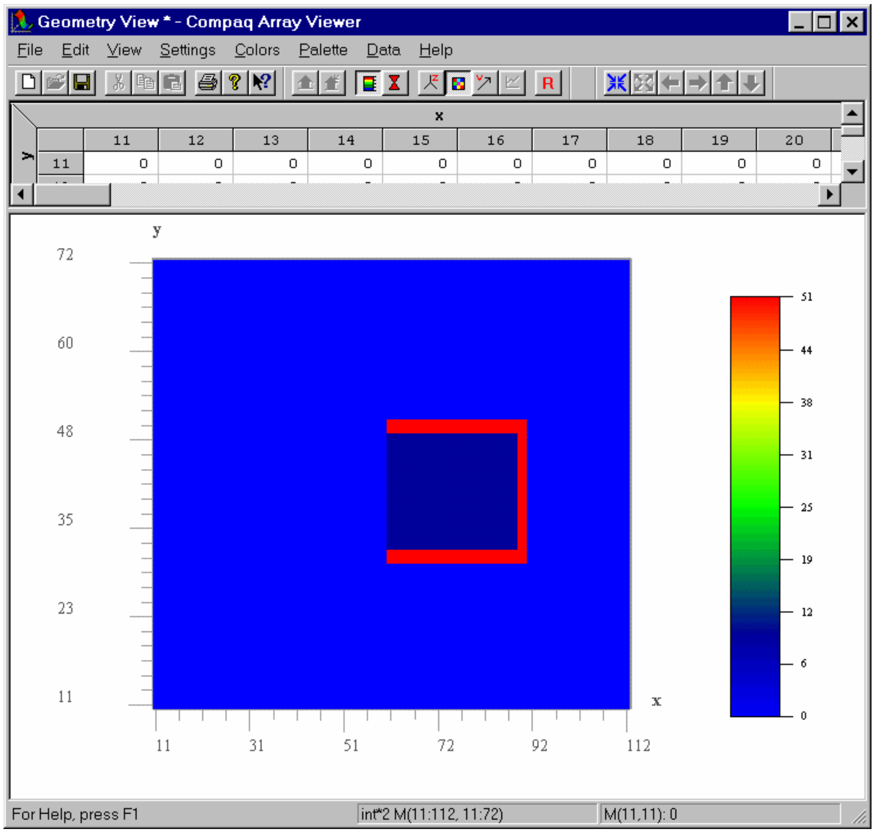
dim: 1

For Help, press F1 | int*2 M(11:112, 11:72) | M(11,11): 0

**Figure 6. The Array Viewer geometry view window of the program *tmpml* showing a 3D schematic representation of the geometry.**

One should note that the indexing scheme shown on the *x* and *y* axes includes the space required for the PML implementation. Thus, in the example above for which 10 PML layers were used, the indexing scheme begins with 11. The non-free space geometry, which was specified in the input data file to begin at node 50, shows up in the array as beginning at 61. By letting the mouse hover over a point in the image map of Figure 7 one can obtain quick (approximate) information on the particular array index and its value via a "tooltip-type" box, but the resolution is somewhat crude. Exact data can be obtained by using the array data window above the plot. The size of the data window can be increased as necessary.

To continue execution the user must first make the Command Prompt window the active window and then press any key to continue. The geometry view window will then close and execution of the FDTD simulation will begin. Since the Command Prompt window must be the active window to continue execution after a pause, it may be convenient to reduce the font size of the Command Prompt window and allow it to remain as the top window once execution starts. Depending on the value of `IPlotPause` in the input data file *tmpml.vwc*, execution may continue until completion, or it may pause after a certain number of time steps. If execution pauses, the Command Prompt window will appear similar to that shown in Figure 8.

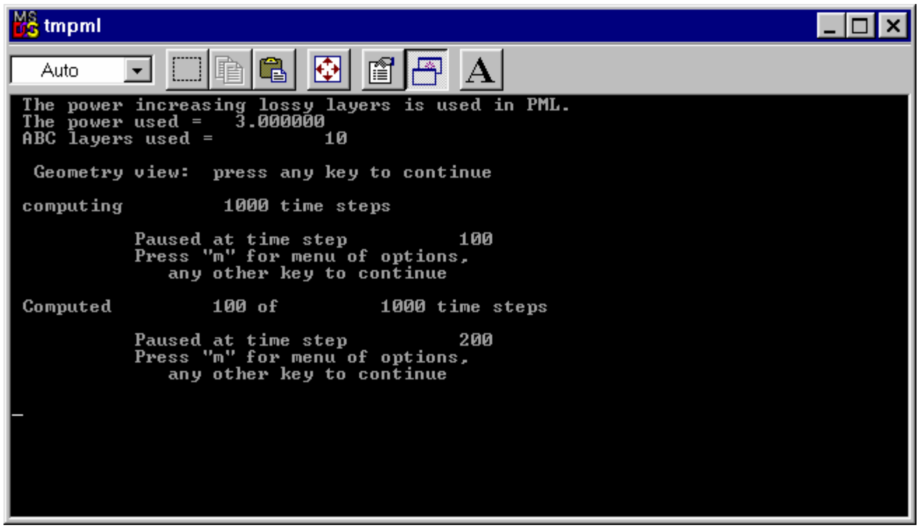


| y \ x | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 7. The Array Viewer geometry view window of the program *tmpml* showing an image map schematic representation of the geometry.**

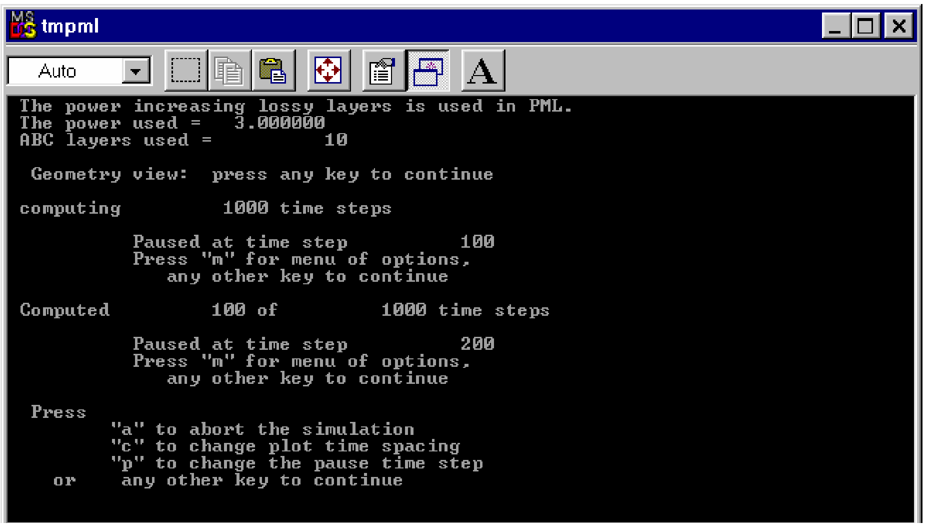
In the window shown in Figure 8 execution has been paused once at 100 time steps, then continued, and has been paused again at 200 time steps. While the program is in this paused state, if the user presses the “m” key a menu will appear as shown in Figure 9.

At this point the user can choose to abort the simulation immediately, to change the plot time spacing (i.e, change the current value of `IPlotStep`), or to change the pause time step spacing (change the current value of `IPlotPause`).



```
The power increasing lossy layers is used in PML.
The power used =    3.000000
ABC layers used =           10

 Geometry view:  press any key to continue

computing         1000 time steps

          Paused at time step          100
          Press "m" for menu of options,
             any other key to continue

Computed         100 of         1000 time steps

          Paused at time step          200
          Press "m" for menu of options,
             any other key to continue

_
```

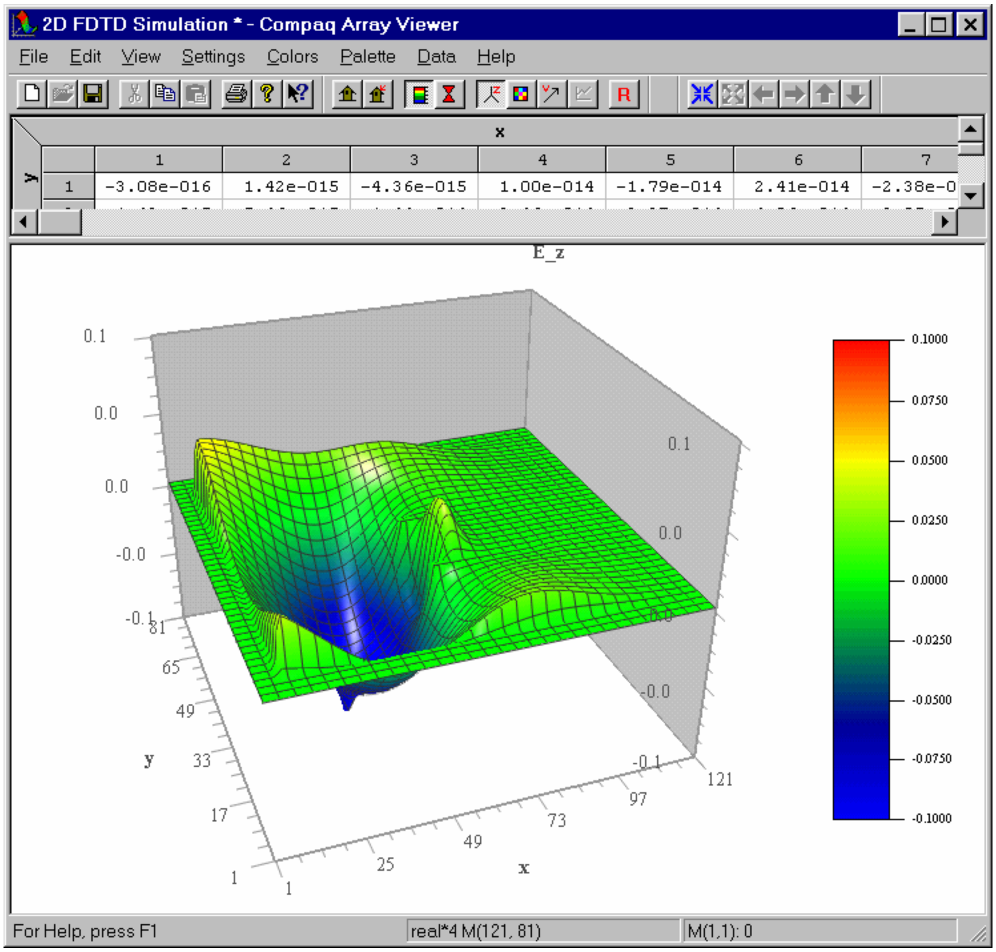
**Figure 8. The *tmpml* Command Prompt window after execution has paused at a time step.**



```
The power increasing lossy layers is used in PML.
The power used =    3.000000
ABC layers used =           10

 Geometry view:  press any key to continue

computing         1000 time steps

          Paused at time step          100
          Press "m" for menu of options,
             any other key to continue

Computed         100 of         1000 time steps

          Paused at time step          200
          Press "m" for menu of options,
             any other key to continue

 Press
        "a" to abort the simulation
        "c" to change plot time spacing
        "p" to change the pause time step
   or    any other key to continue
```

**Figure 9. The *tmpml* Command Prompt window after the user has pressed “m” to get a menu of options.**

For the TM case example data, when execution is paused at time step number 200 the Array Viewer window will appear similar to that shown in Figure 10. Note that the PML region surrounding the computational grid is also displayed in the simulation window.



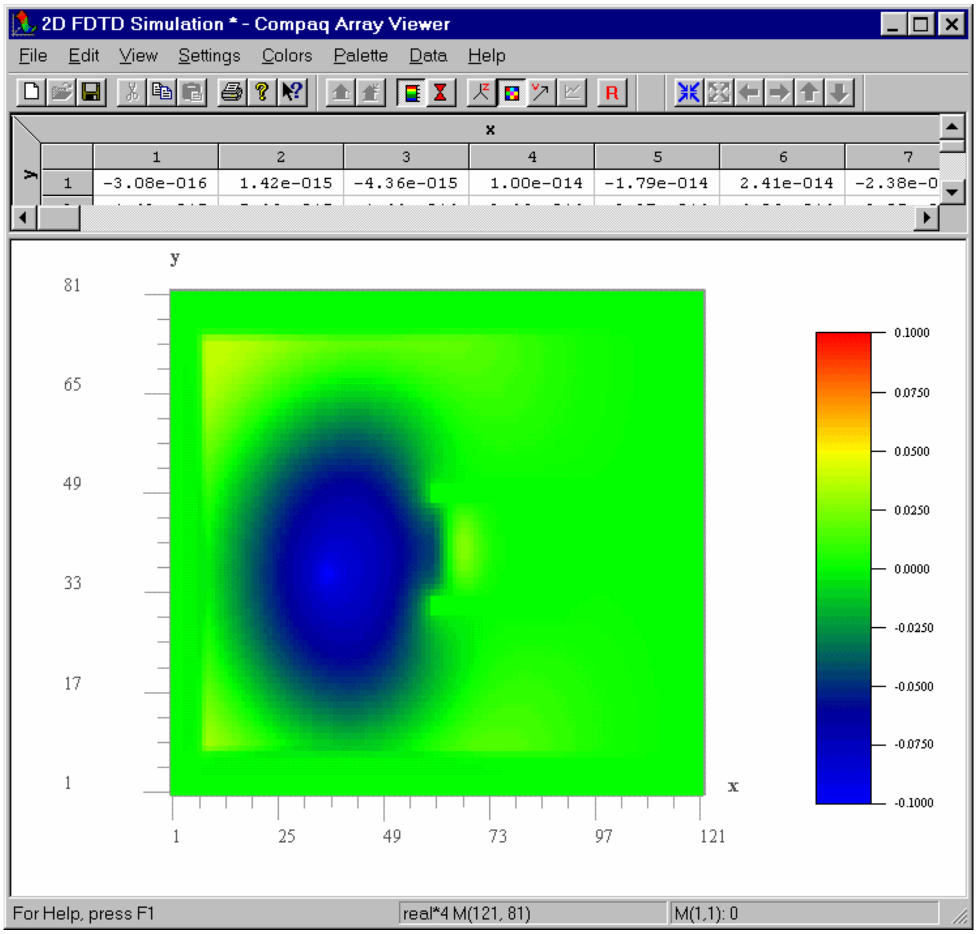
| | | | | x | | | |
|---|---|---|---|---|---|---|---|
| y | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | -3.08e-016 | 1.42e-015 | -4.36e-015 | 1.00e-014 | -1.79e-014 | 2.41e-014 | -2.38e-0 |

For Help, press F1 | real*4 M(121, 81) | M(1,1): 0

**Figure 10. The Array Viewer simulation window of the program *tmpml* showing a 3D view of the electric field distribution.**

As with the geometry window, the user may rotate the view in the simulation window interactively with the mouse to obtain different views. The user may also interact with the menu items and tool bars of the Array Viewer program to zoom in to view a smaller region of space, change the plot scale, change to an image map view of the electric field distribution array as shown in Figure 11, or change other Array Viewer options.

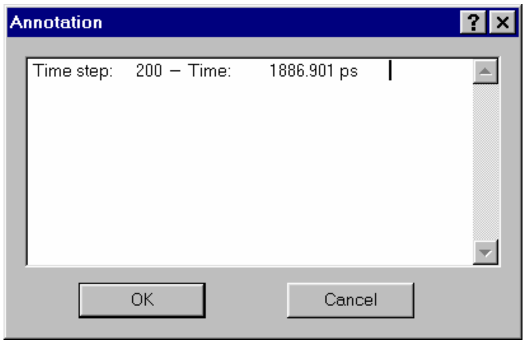
The user can also find the current time step value and the current time value in picoseconds by using the Array View menu. To see this information select Data, then Annotation… The Annotation will appear as shown in Figure 12.


2D FDTD Simulation * - Compaq Array Viewer
File Edit View Settings Colors Palette Data Help
R
y
81
65
49
33
17
1
1
25
49
73
97
121
x
0.1000
0.0750
0.0500
0.0250
0.0000
-0.0250
-0.0500
-0.0750
-0.1000

| y \ x | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | -3.08e-016 | 1.42e-015 | -4.36e-015 | 1.00e-014 | -1.79e-014 | 2.41e-014 | -2.38e-0 |

For Help, press F1 | real*4 M(121, 81) | M(1,1): 0

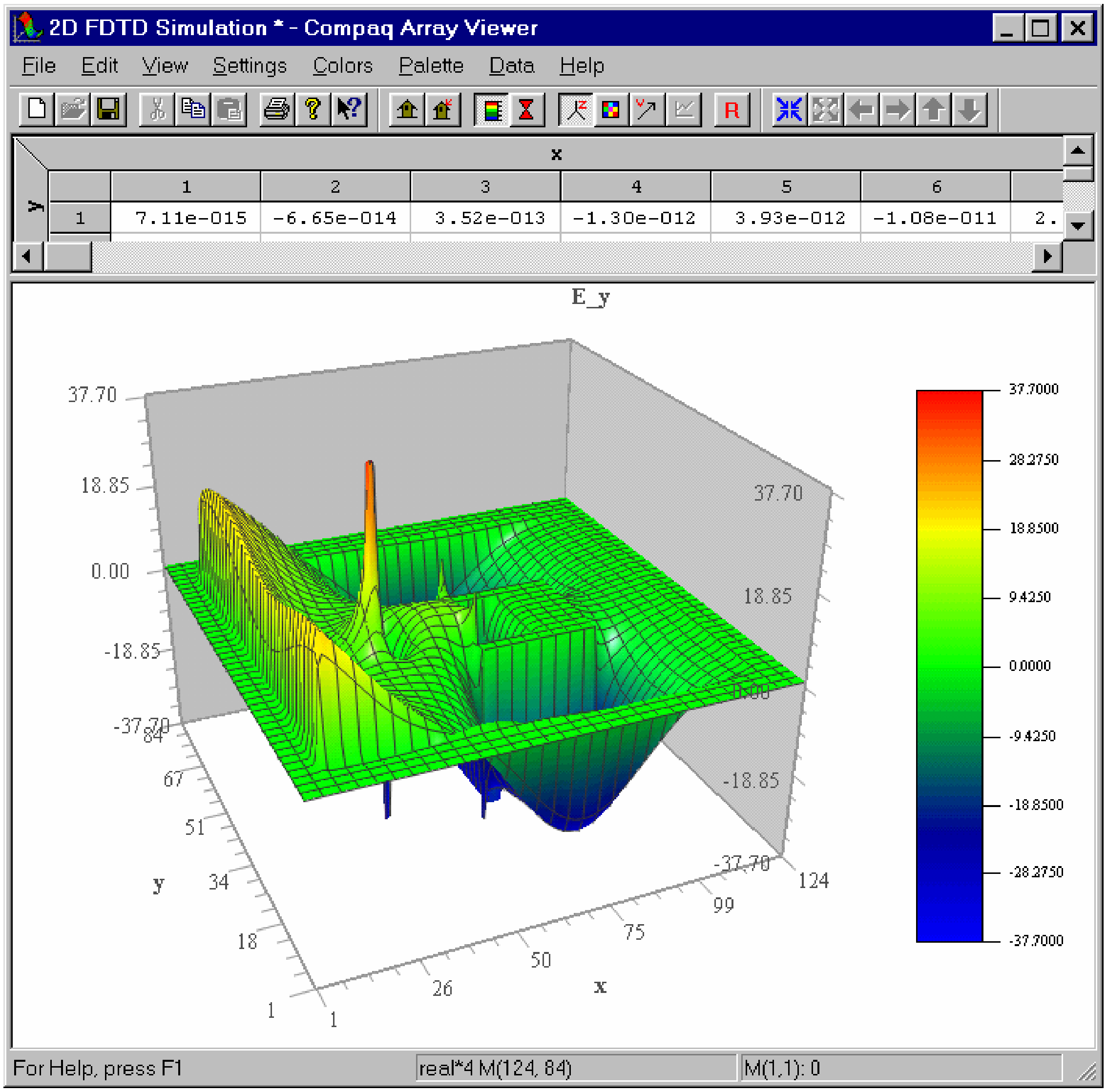
**Figure 11.** The Array Viewer simulation window of the program *tmpml* showing image map view of the electric field distribution.



Time step: 200 – Time: 1886.901 ps

**Figure 12.** The Array Viewer Annotation window.

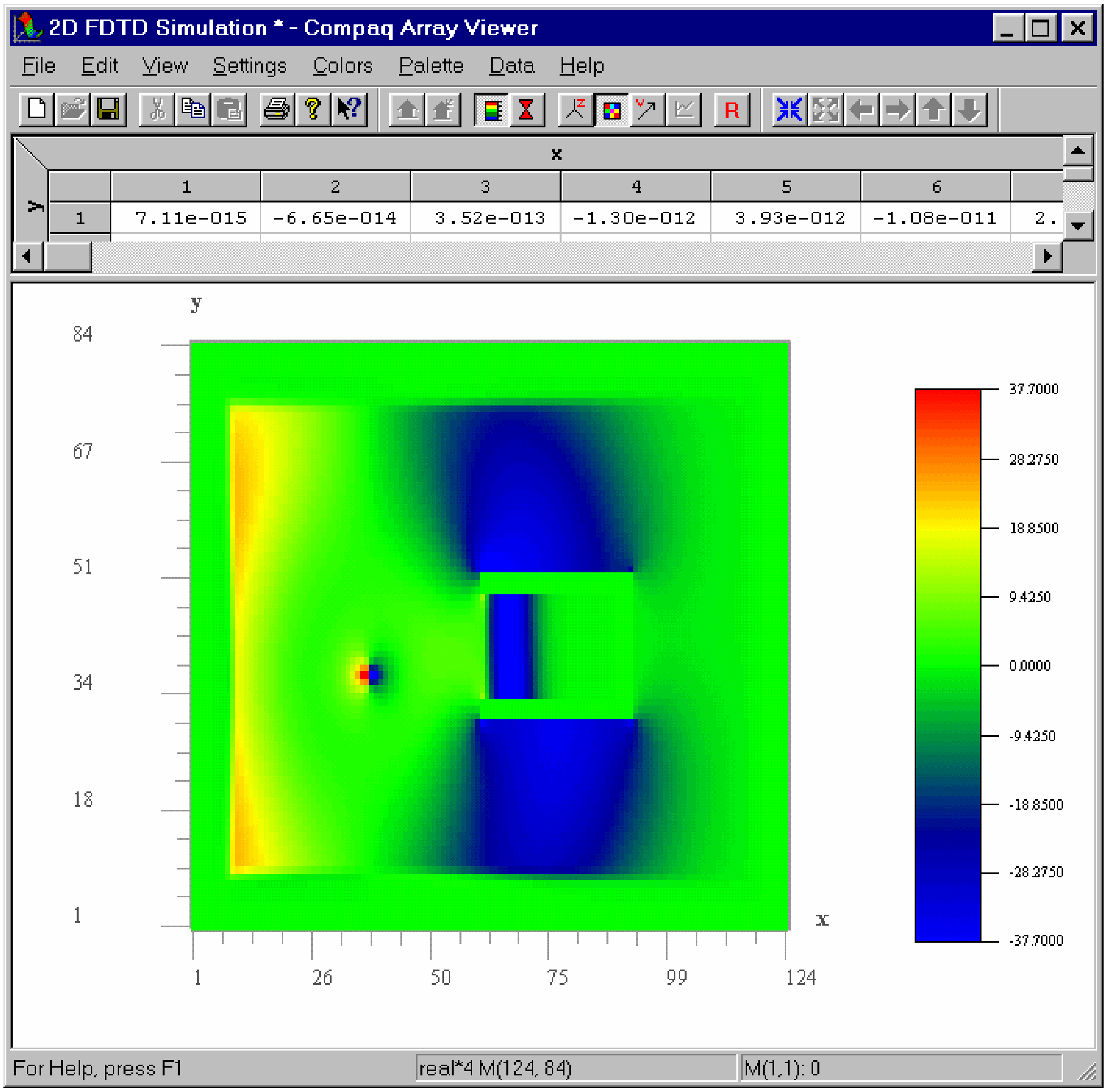
Execution of the TE program *tepml.exe* proceeds in exactly the same manner. Example screens for the TE simulation case are shown in Figures 13 and 14, when execution is paused at time step number 200 for the sample data case.



| y \ x | 1 | 2 | 3 | 4 | 5 | 6 | |
|---|---|---|---|---|---|---|---|
| 1 | 7.11e-015 | -6.65e-014 | 3.52e-013 | -1.30e-012 | 3.93e-012 | -1.08e-011 | 2. |

For Help, press F1 | real*4 M(124, 84) | M(1,1): 0

**Figure 13. The Array Viewer simulation window of the program *tepml* showing a 3D view of the *y* component of the electric field distribution.**



| y \ x | 1 | 2 | 3 | 4 | 5 | 6 | |
|---|---|---|---|---|---|---|---|
| 1 | 7.11e-015 | -6.65e-014 | 3.52e-013 | -1.30e-012 | 3.93e-012 | -1.08e-011 | 2. |

For Help, press F1

real*4 M(124, 84)

M(1,1): 0

**Figure 14.** **The Array Viewer simulation window of the program *tepml* showing image map view of the *y* component of the electric field distribution.**