# Documentation for 2D TM FDTD Code
# tmpml


Allen W. Glisson
aglisson@olemiss.edu

Atef Z. Elsherbeni
atef@olemiss.edu

Dept. of Electrical Engineering
University of Mississippi
University, MS  38677


March 14, 2000

## Introduction

This document briefly describes the use of the 2D Finite Difference Time Domain (FDTD) code "tmpml." This code has been developed primarily for educational use. The program animates time domain scattering by a material geometry excited by a *z*-directed electric current line source. A fairly general scattering geometry can be described by defining various regions with different isotropic constitutive parameters. The program cannot model anisotropic materials in its current form. The perfectly matched layer (PML) absorbing boundary condition is applied at the computational boundaries.

## System Requirements

This code was developed with Compaq Visual Fortran version 6.1 and the Compaq Array Viewer version 1.1. It has been tested under Windows98. The program and its associated example input data files comprise the following five files:

tmpml.exe
tmpml.dim
tmpml.geo
tmpml.vwc
tmpml.ind

In addition, if the machine on which the program is 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 have the Compaq Array Viewer installed, he should verify that the following DLL file aview110.dll is in the Windows System directory. This file is included in this package and should be placed either in the Windows System directory or in the same directory as the FDTD program if it is not already available in the Windows System directory.

## Usage

The program "tmpml.exe" is a "Fortran Console Application" which runs in a DOS box under Windows. The user interacts with the FDTD program through keyboard input to the DOS box. 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 program "tmpml.exe" can be run in any of the usual methods for executing Windows programs.

## Input Data Description

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

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

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

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

`Ngx` and `Ngy` are the numbers of grid points in the *x* and *y* dimensions of the computational grid. These grid points are the points at which the *z* component of the electric field is computed. All non-free-space materials of the scattering geometry reside within this region. There is always one layer of free-space material surrounding the computational grid in addition to the PML region. `NPMLlayers` is the number of PML layers to be used to absorb the outgoing waves. `PMLpower` is the power law to be applied in the PML region. `Nax` and `Nay` are the numbers of additional free-space layers to include between the computational grid and the PML region.

`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 following form (free format):

```
IGeomDraw
IFieldDraw
IPlotStep
IPlotPause
PlotPeak
```

**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 the *z* component of the electric field. 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 DOS 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.

`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 following form (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 $e_r$ (`epsr`), the relative permeability $m_r$ (`mur`), the electric conductivity $s^e$ (`sigmae`), and the magnetic conductivity $s^m$ (`sigmam`). Thus, the first `NumMatDefs+1` lines of this data file represent material definition data available to the program.

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`. Note that these quantities represent *node* values where the *z* component of the electric field is computed. Thus, in this code, the materials are specified to fill the spatial region with the corners defined by the four specified nodes.

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 (the computation grid ranges must still be present as suggested by the data file form shown above, but 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 following form (free format):

```
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.***

The descriptions of the input variables in this file follow:

`Nstop` Final time step number to use.

`JzAmp` Excitation current (Real, signed) amplitude.

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

`Nsourcex, Nsourcey` Node indices within the computational grid in the $x$ and $y$ directions of the electric current line source $J_z$.

`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 $t = M \max(\Delta x, \Delta y)/(2c\sqrt{3})$. A value must appear in the data file for this variable in all cases, but is used only for the "Gaussian" excitation.

`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` `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`).

## Running the Code

When the code begins execution, a DOS window similar to that shown in Fig. 5 will appear. If `IGeomDraw` has been set to one in the data file *tmpml.vwc*, the geometry view window will also appear (Fig. 6) and will be the active window. In the geometry window in Fig. 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).

With 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 Fig. 7, or change other Array Viewer options.


**Figure 5. The *tmpml* DOS window.**

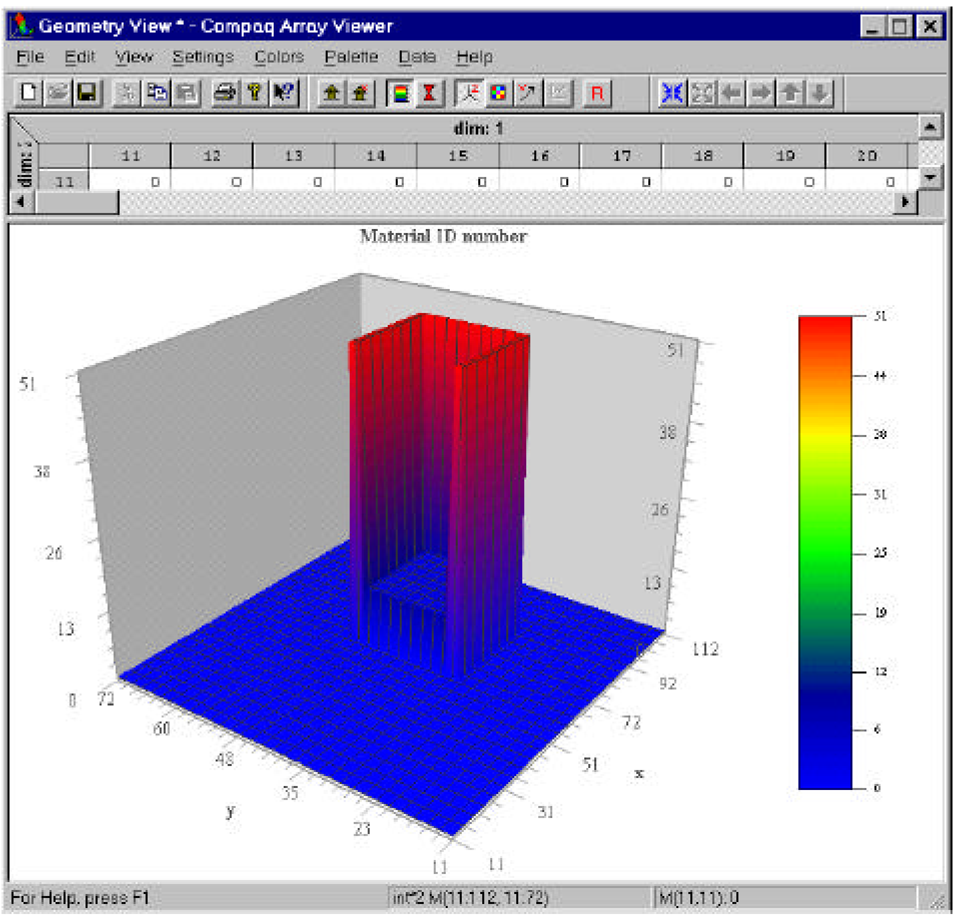


**Figure 6. The Array Viewer geometry view window of the program *tmpml* showing a 3D schematic representation of the geometry.**

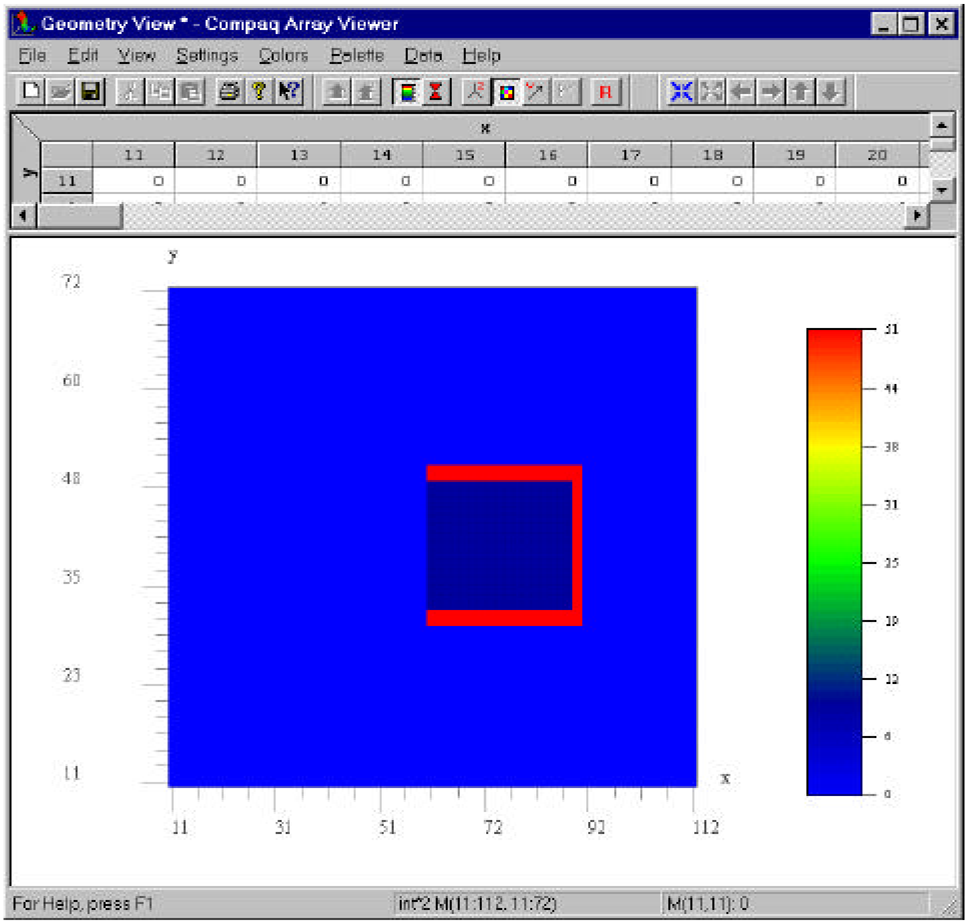


**Figure 7. The Array Viewer geometry view window of the program *tmpml* showing an image map schematic representation of the geometry**

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 Fig. 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 DOS 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 DOS window must be open to continue execution after a pause, it may be convenient to reduce the font size of the DOS 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 DOS window will appear similar to that shown in Fig. 8.

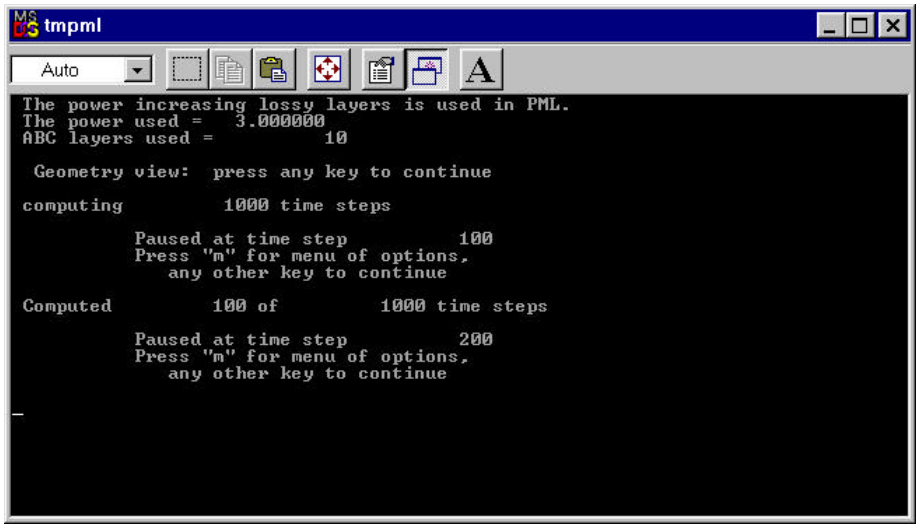


**Figure 8. The *tmpml* DOS window after execution has paused at a time step.**

In the window shown in Fig. 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 Fig. 9.

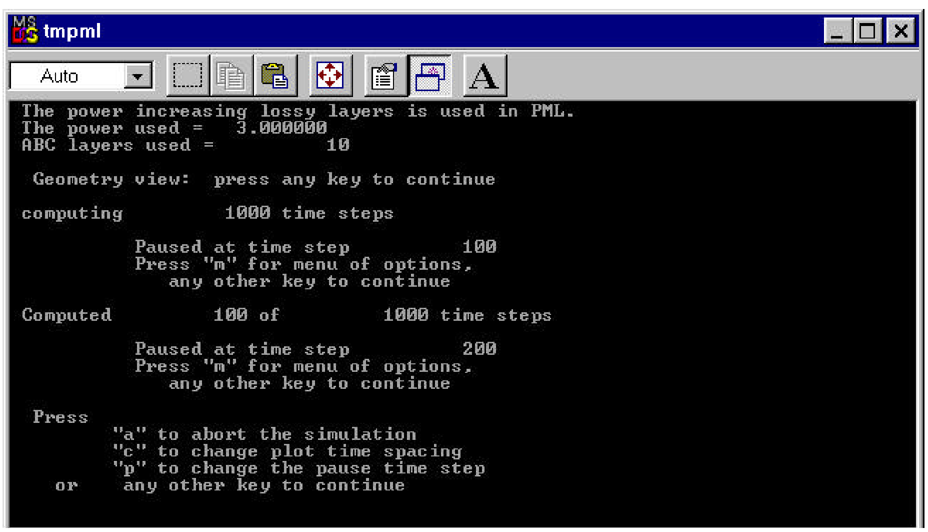


**Figure 9. The *tmpml* DOS window after the user has pressed “m” to get a menu of options.**

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`).

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

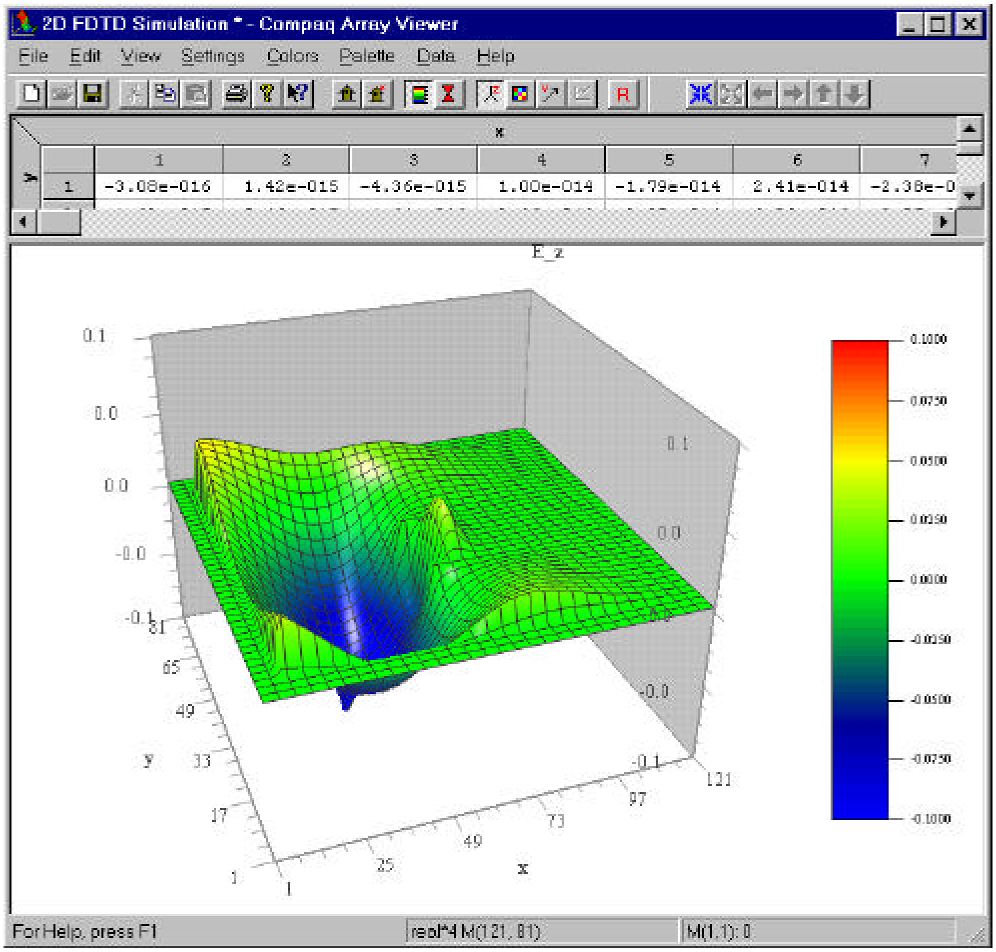


**Figure 10. The Array Viewer simulation window of the program *tmpml* showing a 3D view of the electric field distribution.**

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 Fig. 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 Fig. 12.

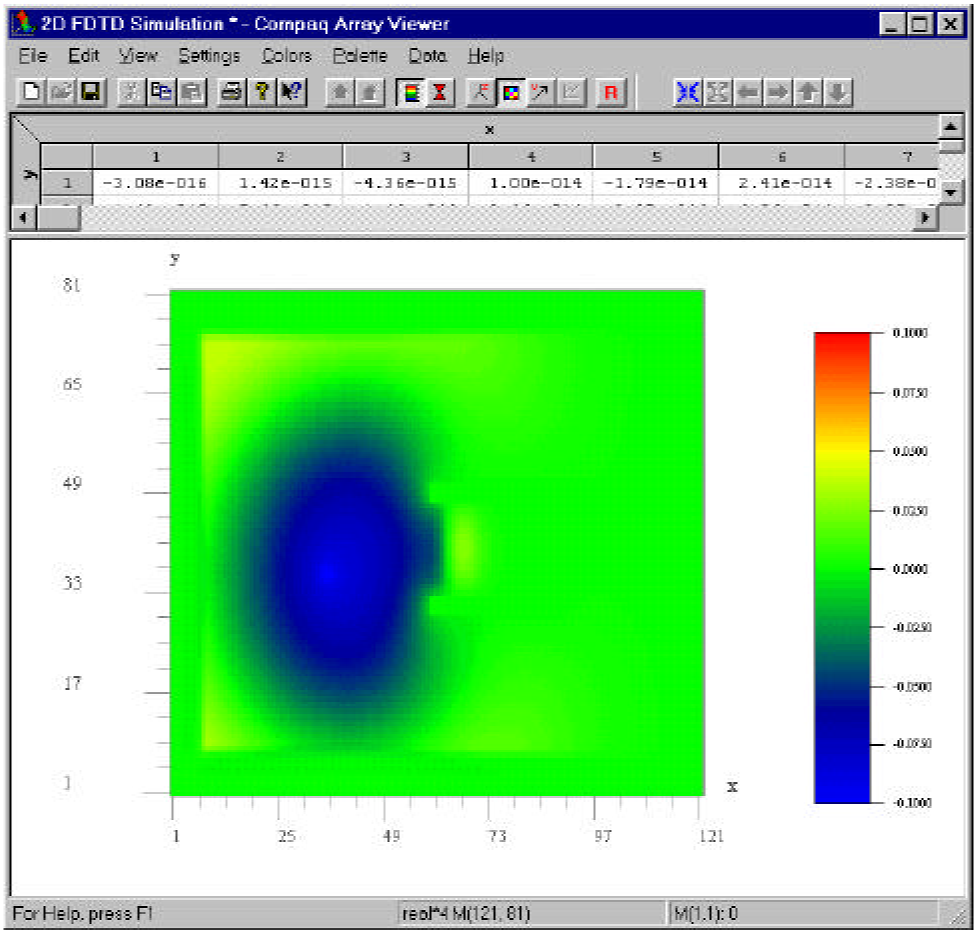


**Figure 11. The Array Viewer simulation window of the program *tmpml* showing image map view of the electric field distribution.**

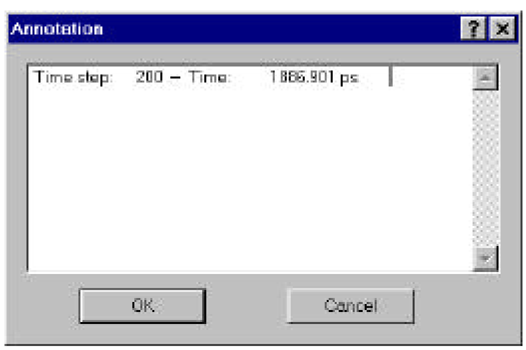


**Figure 12. The Array Viewer Annotation window.**