

Running GGI_TLM on Windows.

GGI_TLM may be run on windows machines by using cygwin. Cygwin creates a unix type environment in which GGI_TLM can be compiled and run. This document describes the installation of paraview (used for result visualisation), cygwin and the download, setup and compilation of GGI_TLM. This document should be sufficient to get the system running and give a brief check that it is working correctly. The user is referred to the documentation which comes along with the installation (user guide, theory manual, test case documentation) for further information.

Paraview installation

Install paraview for windows from the website:

(Note that Paraview version 4.0 is recommended as later versions may not support the file formats used by GGI_TLM.)

www.paraview.org

and follow the download instructions.

Cygwin installation

Go to the website www.cygwin.com

Under the heading Current Cygwin DLL version

Download the cygwin setup for your machine (32 or 64 bit as appropriate) i.e. click setup-x86.exe or setup-x86_64.exe as appropriate (the following notes are based on 64bit installation but I guess it shouldn't be too different for 32 bit.)

Save the file

Double click on the file to run it. (i.e. run the cygwin setup executable)

Click next at the copyright page

Choose a download source:

(e.g. Install from Internet)

Then click next

Choose Root directory:

(e.g. C:\cygwin64)

Select Install for all users

Then click next

Select Local Package Directory

(e.g. C:\Program Files (x86)\Mozilla Firefox)

Then click next

Select Your Internet Connection as appropriate

(e.g. Direct Connection)

Then click next

Choose a download Site

(e.g. cygwin.mirror.uk.sargasso.net works fine in the UK)

Then click next

Select Packages (i.e. what to install) Default is a minimal install and we need some extra stuff so add the following (by clicking where it says 'default' to change to 'install'):

Devel

X11

In the following click the + sign to expand the top level list (e.g. Editors) , find the package in the list (e.g. nedit), then click where it says 'skip' this then changes to the version number to be installed.

Editors->nedit

Graphics->gnuplot

Libs->libopenmpi-devel

Libs->libopenmpi-fh2

Then click next

Resolving dependencies

Ensure that the 'select required packages' box is ticked

Then click next

(I got an error here but things did proceed OK inspite of this)

The Cygwin setup proceeds – this takes quite a long time...

Postinstall script Errors:

Click next

Create Icons

Ensure that the 'Create icon on desktop' box is ticked

Click Finish

An icon 'Cygwin64 Terminal' appears on the desktop

Run Xlaunch from the programs menu

(Programs->Cygwin-X->Xlaunch)

Select Display Settings:

Select 'Multiple windows'

Click next

Select how to start clients

Click 'start no client'

Click next

Extra settings

Tick clipboard and Native OpenGL

Click next

Configuration complete

Click finish.

Start a cygwin terminal by clicking the icon on desktop.

Find the name of your cygwin home directory with the command

pwd

(The result will be something like /home/user)

GGI_TLM

Goto the website

www.github.com/ggiemr/GGI_TLM

click 'download ZIP' button on the right hand side.

save (or copy) the file (GGI_TLM-master.zip) to the cygwin home directory (From windows it will look something like C:\cygwin64\home\user)

Start a cygwin terminal (icon on desktop)

Look at the files in your home directory with the command

ls

This should include the file GGI_TLM-master.zip

Expand the zip file with the command

unzip GGI_TLM-master.zip

You should now have the following directory in you home directory (use **ls**)

GGI_TLM-master

In your home directory, edit the .bash_profile (command **nedit .bash_profile**) file

nedit .bash_profile &

(note: following a command with & runs the command in the background and returns control to the terminal. If you get an error: 'Error: Can't open display' then use the following command: **export DISPLAY=:0** and try nedit again.)

add the following lines at the end of the .bash_profile file:

PATH="\$HOME}/GGI_TLM-master/GGI_TLM/bin:.\$PATH"

export DISPLAY=:0

Then save the file and exit nedit.

(this only needs to be done once when you set up the system, once you have edited the file, exit the cygwin terminal and restart it)

The next stage is to set things up to make GGI_TLM. First go into the main GGI_TLM directory

cd GGI_TLM-master/GGI_TLM

edit the file make_configuration_data with

nedit make_configuration_data &

and set the name of the fortran compiler correctly for cygwin

FC=mpif90

Also set the MPI_INCLUDE directory correctly for cygwin

MPI_INCLUDE=/usr/include

The compilation flags should not need to be changed

edit the file make_run_configuration_data with

nedit run_configuration_data &

and set the MPI_RUN_COMMAND for cygwin

MPI_RUN_COMMAND="mpirun -np "

The SEQ_RUN_COMMAND should be "" but this shouldn't require changing.

Go into the source code directory

cd SRC

and make both sequential and mpi versions of GGI_TLM with the commands:

make clean

make all

This process may take a few minutes...

Once the system is made correctly you can try to run some of the test cases.

Go into the TEST_DATA directory

cd GGI_TLM-master/GGI_TLM/TEST_DATA

run the DIPOLE test case for example with

run_automatic_test run_seq DIPOLE

plot the results of the simulations against the reference results to check everything is OK with

runAutomaticTest plot_ref DIPOLE

A gnuplot window should appear with plots (hit enter in the terminal window to step through the sequence) of time domain electric field, time domain current at the dipole feed point and dipole impedance as a function of frequency.

This test case produces some results which can be viewed with paraview. Start paraview and open files in the DIPOLE directory (the path will be something like

C:\cygwin64\home\user\GGI_TLM-master\GGI_TLM\TEST_DATA\DIPOLE)

For example the file E_field_13MHz.frame...vtk shows an animation of the vector E field at 13MHz on a plane close to the dipole antenna. Other files are available with the dipole geometry, mesh, outer boundary etc. These are described in the documentation.