Using CST with Matlab

Method · January 2019

1 author:

Elena Mikheeva
CRHEA
6 PUBLICATIONS **24** CITATIONS

SEE PROFILE

Running CST Studio simulations with Matlab

This document is to describe how to run a simulation in CST Studio from Matlab: how to pass commands, change parameters, save the result and others.

It can be done by using ActiveX framework (https://en.wikipedia.org/wiki/ActiveX)

It is sufficient to only use few basic ActiveX functions:

`actxserver –` initialization (opening) of the controlled program (CST)

`invoke –` calling the units of controlled program

Examples:

```
cst = actxserver('CSTStudio.Application')
```

Command connects variable “cst” and object “CSTStudio.Application”. In this case “CSTStudio.Application” is a unique name in the ActiveX environment showing which program we are calling.

```
mws = invoke(cst , 'NewMWS')
```

Like that we are sending a command into the variable “cst” connected with the application CST Studio to create a new empty window

```
invoke(mws, 'OpenFile', '<File path>')
```

Sends a command to open a concrete file situated in the “File path” directory in a just opened empty window connected with a variable “mws”

```
solver = invoke(mws, 'Solver')
```

Writes into the variable “solver” a call to the time domain-solver. If we want to call other solvers, we should replace ‘Solver’ with ‘FDSolver’ for frequency domain, ‘EigenmodeSolver’ for eigenmode and so on.

```
invoke(solver, 'start')
```

Starts the solver

```
invoke(mws, 'save')
```

Saves changes

```
invoke(cst, 'quit')
```

Closes the file.

More explanation and examples:

1. General description of using CST through Matlab, building the object, changing units and parameters

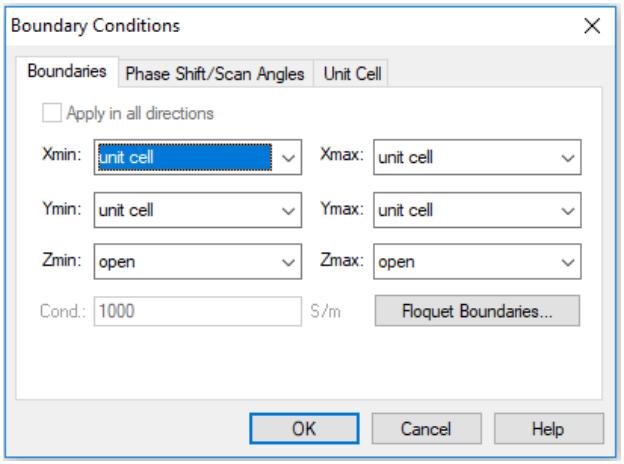
https://habr.com/ru/post/325012/

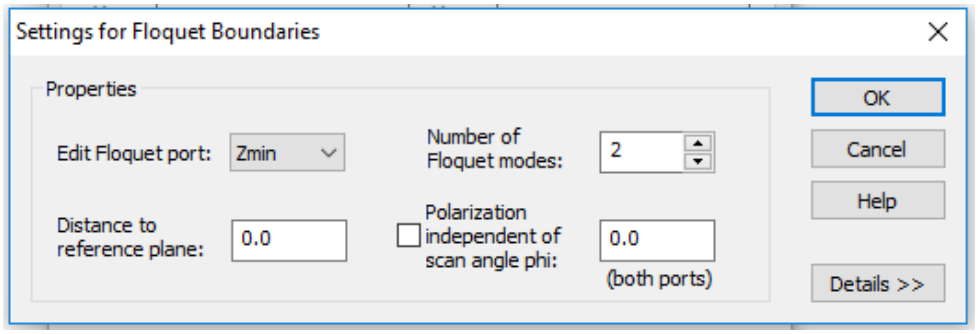
2. Concrete example of antenna simulation (parameters sweep and results treatment) with scripts:

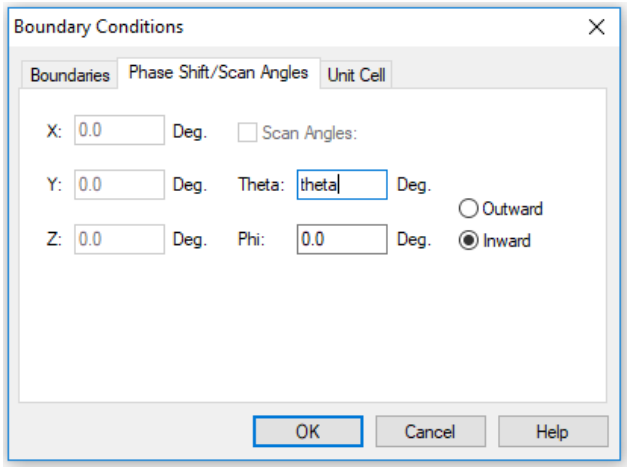
https://habr.com/ru/post/336396/

My example:

Frequency-domain simulation of the periodic structure with the unit cell boundary conditions. I am considering 2 Floquet modes corresponding to TM and TE polarizations and making simulations at oblique incidence:







```
%running an oblique incidence simulations in CST with a periodic
%structure

clear all; clc;
close all;

cst = actxserver('CSTStudio.Application');
mws = cst.invoke('NewMWS');
plot1d = mws.invoke('Plot1D');
export = mws.invoke('ASCIIExport');
mws.invoke('OpenFile','file_path\file_name.cst');

N = 10;
theta=linspace(0,45,N);

for ii = 1:N
    mws.invoke('StoreDoubleParameter','theta', theta(ii)); %send into
parameter theta a new value
    mws.invoke('Rebuild'); %rebuild a structure

    hSolver = invoke(mws,'FDSolver');
    invoke(hSolver,'Start');

    str_theta = num2str(theta(ii));
    f_name = strcat('file_path\file_name\',str_theta,'deg.txt'); %create a
name

    %saving the result, absolute value, TM-polarization
    mws.invoke('SelectTreeItem','1D Results\S-Parameters\SZmin(1),Zmax(1)');
%select results
    plot1d.invoke('PlotView','magnitude'); %switch to the absolute value, can
be 'phase' or others
    export.invoke('Reset');
    export.invoke('FileName',f_name);
    export.invoke('Mode','FixedNumber');
    export.invoke('Step','1001'); %number of points can be changed in CST in
simulation, setup solver, properties
    export.invoke('Execute')

    t = h_remove(f_name); %using a script from the link 2 to remove first 2
lines in a saved file
    A(:,ii)= t(:,2)'; %saving results for each value into a matrix
end

%saving a matrix
fid = fopen('A.txt','wt');
for ii = 1:size(A,1)
    fprintf(fid,'%g\t',A(ii,:));
    fprintf(fid,'\n');
end
fclose(fid)
```

More information on possible commands can be found in CST help: