DEVSIM BJT Example

Juan E. Sanchez
juan.sanchez@devsim.com

April 23, 2024

1 Introduction

This package includes the examples for the article "Semiconductor Device Simulation Using DEVSIM". In this example, the meshing, modeling, simulation, and visualization for a bipolar junction transistor (BJT) is developed. DEVSIM is an open source simulation software for technology computer-aided design (TCAD) and is developed by DEVSIM LLC. It uses a generalized partial-differential equation (PDE) approach to perform semiconductor device simulation on a mesh. The software and documentation is available from http://www.devsim.org.

In addition to DEVSIM, the following software packages are used to perform meshing, analysis, and visualization of results.

Name	Description	Website	License[a]
Gmsh	Mesh Generator	http://geuz.org/gmsh	GPL
matplotlib	Python 2D Plotting Library	http://matplotlib.org	matplotlib
NumPy	Python Scientific Computing	http://numpy.org	BSD
Python	Scripting Language	http://python.org	PSF
VisIt	Visualization Tool	http://visit.llnl.gov	BSD

2 Running the Examples

Here are some of the files in the package used for simulation.

`bjt.geo`	Mesh description for Gmsh
`bjt.msh`	Resulting gmsh mesh
`initial_guess.py`	Creates initial guess from Potential only simulation
`refinement.py`	Sets up E-field based refinements for creating background mesh
`netdoping.py`	Specifies analytical doping profile
`bjt_dd.py`	Creates the zero bias drift diffusion solution for later sweeps
`bjt_refine.py`	Runs DEVSIM to create a background mesh
`bjt_bgmesh.pos`	Background mesh generated by DEVSIM for refinement using Gmsh
`physics/`	subdirectory containing physics files used in simulation.

2.1 Meshing and Refinement

The file `bjt.geo` contains the initial mesh specification for the bjt structure. This file is run through Gmsh in order to create a triangular mesh for use in DEVSIM. The resulting mesh file is called `bjt.msh`. In order to create a mesh suitable for devsim, the `bjt_refine.py` script is run to create a background mesh with element sizes appropriate for simulation. The background mesh is then used with the original mesh specification to create a refined mesh. This procedure is repeated until the mesh is sufficiently refined for use in DEVSIM.

The steps are:

```
gmsh -format msh2 -2 bjt.geo
python bjt_refine.py
gmsh -format msh2 -2 bjt.geo -bgm ./bjt_bgmesh.pos
python bjt_refine.py
gmsh -format msh2 -2 bjt.geo -bgm ./bjt_bgmesh.pos
python bjt_refine.py
gmsh -format msh2 -2 bjt.geo -bgm ./bjt_bgmesh.pos
python bjt_refine.py
gmsh -format msh2 -2 bjt.geo -bgm ./bjt_bgmesh.pos
python bjt_refine.py
```

The resulting mesh from each DEVSIM run can be visualized by running Visit.

```
visit bjt_refine.tec
```

2.2 Zero bias drift diffusion solution

When creating a new device, it is necessary to create initial zero bias solution for all the subsequent sweeps.

```
python bjt_dd.py
```

This creates `bjt_dd_0.msh` file that is read in the next section.

2.3 Simulation

The dc and ac sweeps used in the publication are listed in `simsbatch.txt`. These simulations can be run in sequence or in parallel.

V_c sweep

For a given value of V_b, sweep V_c from 0 to 1.5 V.

```
python bjt_circuit2.py 0.1 &> data/vb2_0.1.out
```

V_b sweep

For a given value of V_c, sweep V_b from 0 to 1.0 V.

```
python bjt_circuit3.py 0.0 &> data/vc_0.0.out
```

V_e sweep

For a given value of V_c, sweep V_e from 0 to -1.0 V.

```
python bjt_circuit4.py 0.0 &> data/ve_0.0.out
```

Small-signal ac sweep

For a given value of V_c, sweep V_e from 0 to -1.0 V. Do a small signal frequency sweep from fmin to fmax with given points per decade.

```
python bjt_circuit5.py 0.0 1e3 1e11 3 &> data/ssac_0.0.out
```

2.4 Visualization

The `data/` directory contains scripts used to generate the plots used for publication. A script was written to collect the data from the simulations to create plots using `matplotlib`. This script is in the `data/` directory and is called `prep.sh`.

`ft.py`	Small-signal ft simulation
`gummel.py`	Ic, Ib versus Vbe.
`ic_vec.py`	Ic versus Vce.