SDS 半导体模拟器

Semiconductor Device Simulator

操作手册

版本号：1.0

版权所有：陈秋松

目录

第一章、 目标和背景

本软件旨在利用数值计算方法模拟半导体、铁电等物理特性的电子器件，并利用模拟结果分析器件中的物理过程，包含电流电压关系、半导体缺陷态生成复合过程、铁电材料内部极化过程等。

本软件核心求解工具为 DEVSIM[1]，DEVSIM 是一个开源的半导体模拟软件，本软件是在 DEVSIM 的基础上进行二次开发，并针对特定类型的物理器件开发了专门的解决办法。

DEVSIM 基于有限体积法构建半导体中电场的高斯定理以及载流子的连续性方程的迭代公式，并使用牛顿迭代法求解，其迭代公式主要基于符号函数方式构建、因此可以比较便捷地将电子器件中各种物理模型的公式写成对应符号函数的形式，十分利于复杂物理模型的实施和求解。理解 DEVSIM 中的基本概念和使用方法是使用本软件的基础，DEVSIM 具体数学原理和使用说明可以参阅网站 https://devsim.net/

[1] DEVSIM LLC: DEVSIM TCAD semiconductor device simulator. Available: https://devsim.org

第二章、 软件组成模块

本软件目前包含以下功能模块（每个模块均为 Python 扩展程序库的形式）：

1) 创建、删除、核对 DEVSIM 模型
   QSmodel_create.py
2) 创建硅半导体物理关系模型
   QSsimple_physics.py
3) 创建缺陷态物理关系模型
   QSExtraCharge.py
4) 创建铁电材料关系模型
   QSFerro.py
5) 创建肖特基电极关系模型
   QSSchottkyContact.py
6) 执行器件电压扫描
   QSPlotSweep.py
7) 创建一维和二维二极管模拟电子器件
   QSdiode_common.py

第三章、 安装步骤

本软件的使用平台与 DEVSIM 完全一致，使用 Python 作为具体执行平台，因此可以在 Linux、MacOS 以及 Windows 操作系统上使用，推荐使用 Linux 上的 Conda 或者 MiniConda 作为 Python 的配置环境。

除 Devsim 外，本软件的具体使用还需要配合下列开源软件完成特定的前期和后期工作：

1. 多维网格生成软件：Gmsh [https://gmsh.info/
2. 数据可视化和图像生成软件：VisIt [https://www.visitusers.org/

下面将以 Ubuntu Linux 操作系统和 MiniConda 为背景介绍具体安装步骤

1. 安装 MiniConda
   在 MiniConda 网页[https://docs.conda.io/projects/miniconda/en/latest/#]的说明，并根据硬件平台下载后安装，或者通过命令行直接安装
2. 安装 Devsim
   在 Devsim 网页[https://devsim.org/introduction.html#download] 的说明，并根据硬件平台下载后安装
3. 安装 Gmsh
   在 linux 系统命令行中执行下列命令安装 Gmsh
   `sudo apt install gmsh`
4. 安装 VisIt
   在 VisIt 网页[https://www.visitusers.org/的说明，并根据硬件平台下载后安装
5. 必要的 Python 扩展程序库
   本软件必备的 Python 扩展程序库主要包含：mkl, numpy, matplotlib, pandas, scipy, sympy 等，可在 linux 系统命令行中执行下列命令安装：
   `conda install mkl numpy matplotlib pandas scipy sympy`
6. 使用本软件模块
   建议将本软件的所有扩展程序库文件放置于同一路径下，并使用下面 python 语句导入：

```
from devsim import *
import sys, devsim
sys.path.append('[存放路径]')
from [扩展程序库名] import *
```

第四章、 模块-函数功能介绍

1) QSmodel_create.py

本模块用于创建、删除、核对 DEVSIM 中各种模型

CreateSolution(device, region, name)

创建网格节点变量，可用于存储节点数据，或用作有限体积法求解变量

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

name：（字符串）所创建变量名称

CreateNodeModel(device, region, model, expression):

创建网格节点模型（公式）

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所创建节点模型名称

expression：（字符串）所创建节点模型公式，为符号函数形式

CreateNodeModelDerivative(device, region, model, expression, *vars):

创建网格节点模型对多个求解变量的导数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）节点模型量名称

expression：（字符串）所创建节点模型公式，为符号函数形式

*vars：（字符串）求解变量名称

CreateNodeModelAndDerivative(device, region, model, expression, *vars):

创建网格节点模型同时创建网格节点模型对求解变量的导数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所创建节点模型名称

expression：（字符串）所创建节点模型公式，为符号函数形式

*vars：（字符串）求解变量名称

CreateContactNodeModel(device, contact, model, expression):

创建电极上的网格节点模型

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

model：（字符串）所创建变量名称

expression：（字符串）所创建节点模型公式，为符号函数形式

CreateContactNodeModelDerivative(device, contact, model, expression, *vars):

创建电极上的网格节点模型对多个求解变量的导数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所创建变量名称

expression：（字符串）所创建节点模型公式，为符号函数形式

*vars：（字符串）求解变量名称

CreateEdgeModel (device, region, model, expression):

创建网格边模型

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所创建边模型名称

expression：（字符串）所创建边模型公式，为符号函数形式

CreateEdgeModelDerivatives(device, region, model, expression, variable):

创建网格边模型对求解变量的导数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所创建边模型名称

expression：（字符串）所创建边模型公式，为符号函数形式

variable：（字符串）求解变量名称

CreateEdgeModelAndDerivatives(device, region, model, expression, *vars):

创建网格边模型同时创建网格边模型对多个求解变量的导数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所创建边模型名称

expression：（字符串）所创建边模型公式，为符号函数形式

*vars：（字符串）多个求解变量名称

CreateEdgeFromSqrtNode(device, region, model, *vars):

利用网格边两端节点变量乘积的平方根的创建边模型，同时创建网格边模型对多个求解变量的导数，创建边模型的名字为："Sqrt”+ model

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所引用节点变量名称

*vars：（字符串）求解变量名称

CreateContactEdgeModel(device, contact, model, expression):

创建电极上的网格边模型

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

model：（字符串）所创建边模型名称

expression：（字符串）所创建边模型公式，为符号函数形式

CreateContactEdgeModelDerivative(device, contact, model, expression, variable):

创建电极上的网格边模型对求解变量的导数

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

model：（字符串）所创建边模型名称

expression：（字符串）所创建边模型公式，为符号函数形式

variable：（字符串）求解变量名称

CreateInterfaceModel(device, interface, model, expression):

创建界面模型公式

参数含义：

device：（字符串）器件名称

interface：（字符串）界面名称

model：（字符串）所创建界面模型名称

expression：（字符串）所创建边模型公式，为符号函数形式

CreateContinuousInterfaceModel(device, interface, variable):

创建连续性界面模型公式，使同一节点变量在界面上连续变化，创建界面模型的名字为："continuous"+ variable

参数含义：

device：（字符串）器件名称

interface：（字符串）界面名称

variable：（字符串）所使用节点变量名称

InEdgeModelList(device, region, model):

核对指定区域中是否存在特定边模型，返回布尔值

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所核对的边模型

InNodeModelList(device, region, model):

核对指定区域中是否存在特定节点模型，返回布尔值

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所核对的节点模型名称

InElementModelList(device, region, model):

核对指定区域中是否存在特定对偶元边模型，返回布尔值

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所核对的对偶元边模型名称

InSolutionList(device, region, solution):

核对指定区域中是否存在特定节点变量，返回布尔值

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所核对的节点变量名称

InContactList(device, contact):

核对指定器件中是否存在特定的电极，返回布尔值

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所核对的电极名称

InParameterList(device, Parameter, region=None)

核对指定器件的特定区域中是否存在特定的参数，返回布尔值

参数含义：

device：（字符串）器件名称

Parameter：（字符串）区域名称

region：（字符串）区域名称，如不指定区域，则在公共参数中查找

EnsureEdgeFromNodeModelExists(device, region, nodemodel):

核对指定器件的特定区域中是否存在特定的节点模型，如不存在则报错，然后利用此节点模型创建两个边模型，边模型名分别为：nodemodel +“@n0”以及 nodemodel +“@n1”

参数含义：

device：（字符串）器件名称

Parameter：（字符串）区域名称

nodemodel：（字符串）节点模型名称

CreateDimensionLable(device):

创建器件参数"DimList"用于保存坐标标签，一维结构为("x",)，二维结构为("x","y")，三维结构为("x","y","z")

参数含义：

device：（字符串）器件名称

CreateElementModel2d(device, region, model, expression):

创建网格对偶元边模型

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所创建边模型名称

expression：（字符串）所创建边模型公式，为符号函数形式

CreateElementModelDerivative2d(device, region, model_name, expression, *args):

创建网格对偶元边模型对多个求解变量的导数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所创建边模型名称

expression：（字符串）所创建边模型公式，为符号函数形式

*args：（字符串）多个求解变量名称

CreateElementEdgeFromSqrtNode(device, region, model, *vars):

利用网格边两端节点变量乘积的平方根的创建对偶元边模型，同时创建对偶元边模型对多个求解变量的导数，创建边模型的名字为："Sqrt”+ model

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

model：（字符串）所引用节点变量名称

*args：（字符串）多个求解变量名称

DeleteEdgeModelAndDerivatives(device, region, name):

删除指定区域中的特定边模型及其对所有节点变量的导数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

name：（字符串）特定边模型名称

DeleteNodeModelAndDerivatives(device, region, name):

删除指定区域中的特节点模型及其对所有节点变量的导数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

name：（字符串）特定边模型名称

DeleteNodeModelAndDerivates(device, region, name)

删除节点模型及其导数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

name：（字符串）节点模型名称

2) QSsimple_physics.py

以基于玻尔兹曼近似的硅半导体物理关系，创建器件模拟的基本物理模型

InDisplayHost():

核对主机是否属于需要实时显示运行结果的主机，返回布尔值

InNoneCompressHost():

核对当前主机是否属于需要压缩导出文件的主机, 返回布尔值

SetExtendedPrecision(Threads=4,TaskSize=2048):

设置 DEVSIM 运行时使用的线程数和单次计算的元素个数

SetVerbosity(verbose=True):

设置是否以冗长模式实时输出

SetCentiMeterBasicParameters():

以厘米为长度基本单位设置基础物理参数，包括元电荷电量、玻尔兹曼常数、真空介电常数

SetMicroMeterBasicParameters():

以微米为长度基本单位设置基础物理参数，包括元电荷电量、玻尔兹曼常数、真空介电常数

SetOxideParameters(device, region, rPermittivity=1.0):

设置电介质材料的相对介电常数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

rPermittivity：（数值）相对介电常数

SetDopingParameters(device, region, bulk_doping, drain_doping, source_doping):

设置晶体管活性区体、漏极、源极附近参杂浓度

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

bulk_doping：（数值）体参杂浓度，默认为 0

drain_doping：（数值）漏极参杂浓度，默认为 0

source_doping：（数值）源极参杂浓度，默认为 0

SetMicroMeterSiliconParameters(device, region, T, rPermittivity , n_i, nMobility , pMobility, nLifeTime, pLifeTime):

以微米为长度基本单位设置硅材料区域基础物理参数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

T：（数值）器件温度，默认 300K

rPermittivity：（数值）相对介电常数，默认 11

n_i：（数值）本征载流子浓度，默认 1.0e-2

nMobility：（数值）电子迁移率，默认 1e8

pMobility：（数值）空穴迁移率，默认 1e8

nLifeTime：（数值） 电子平均寿命，默认 1e-5s

pLifeTime：（数值） 空穴平均寿命，默认 1e-5s

SetCentiMeterSiliconParameters(device, region, T, rPermittivity , n_i, nMobility , pMobility, nLifeTime, pLifeTime):

以厘米为长度基本单位设置硅材料区域基础物理参数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

T：（数值）器件温度，默认 300K

rPermittivity：（数值）相对介电常数，默认 11

n_i：（数值）本征载流子浓度，默认 1.0e10

nMobility：（数值）电子迁移率，默认 1

pMobility：（数值）空穴迁移率，默认 1

nLifeTime：（数值） 电子平均寿命，默认 1e-5s

pLifeTime：（数值） 空穴平均寿命，默认 1e-5s

SetGaussianDOSCarrierParameters(device, region, TypeName, BulkDensity, Width, Center, Rate0, LocalScale, HDistance):

设置高斯分布能级的基础参数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

TypeName：（字符串）能级类型，可选="GaussianHomo" 或 "GaussianLumo"

BulkDensity：（数值）总态密度

Width：（数值）能级分布宽度

Center：（数值）能级中心位置

Rate0：（数值）跃迁尝试频率

LocalScale：（数值）定域化尺度

HDistance：（数值）载流子在能级内部的平均跃迁距离

CreateGaussianCarriersExchange(device, region, BasicRate, Distance):

设置高斯分布导带和价带之间载流子跃迁模型

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

BasicRate：（数值）跃迁尝试频率

Distance：（数值）平均跃迁距离

CreatePotentialAndFlux(device, region)

设置电势以及电场模型

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

CreateSiliconNetDoping(device, region, bulk_doping)

设置硅材料区域净参杂浓度

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

bulk_doping：（数值）净参杂浓度

CreateSiliconPotentialOnly(device, region):

以平衡态计算器件的电势和载流子浓度，作为初始值使用

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

DeleteIntrinsicNodeModelDerivatives(device, region):

删除初始值计算过程中使用的多余变量和模型

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

CreateSiliconPotentialOnlyContact(device, contact, is_circuit=False, offset=False)

以平衡态设置电极处电势，作为初始值使用

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

is_circuit：（布尔值）是否接入电路

offset：（布尔值）电势是否有错位

CreateSRH(device, region):

设置半导体区的 Shockley-Read-Hall (SRH) 复合模型

CreateECE(device, region, mu_n, Bandtype="Silicon")

设置半导体区的电子电流模型以及公式

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

mu_n,：（数值）迁移率模型

Bandtype：（字符串）能带模型，可选"Gaussian"或"Silicon":

CreateHCE(device, region, mu_p, Bandtype="Silicon")

设置半导体区的空穴电流模型以及公式

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

mu_n,：（数值）迁移率模型

Bandtype：（字符串）能带模型，可选"Gaussian"或"Silicon"

CreatePE(device, region):

设置半导体区的电场高斯定理公式

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

mu_n：（数值）迁移率模型

Bandtype：（字符串）能带模型，可选"Gaussian"或"Silicon"

CreateElectronAndHoleSolutions(device, region, PreValue=True, Bandtype="Silicon")

创建半导体区的电子和空穴求解变量，并根据 PreValue 选择是否从平衡态获取初值

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

PreValue：（数值）迁移率模型

Bandtype：（字符串）能带模型，可选"Silicon"

SetInitialSolutionFromDopping(device, region, Bandtype="Silicon")

根据参杂浓度设置电子和空穴求解变量初值

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

PreValue：（数值）迁移率模型

Bandtype：（字符串）能带模型，可选"Silicon"

CreateSiliconDriftDiffusion(device, region, mu_n", mu_p="mu_p_eff", SRHMolde=True)

创建半导体区的电子和空穴连续性方程

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

mu_n：（数值）电子迁移率模型

mu_p：（数值）空穴迁移率模型

SRHMolde：（布尔值）是否使用 Shockley-Read-Hall (SRH) 复合模型

CreateSiliconDriftDiffusionAtContact(device, contact, is_circuit=False)

创建半导体区电极上的电子和空穴连续性方程，整个电极上的平衡态浓度可根据参杂变化

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

is_circuit：（布尔值）是否接入电路

CreateElectronAndHoleAtContact(device, contact, is_circuit=False)

创建半导体区电极上的电子和空穴连续性方程，整个电极上的平衡态浓度为均匀值

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

is_circuit：（布尔值）是否接入电路

CreateOxidePotentialOnly(device, region, update_type="ault"):

创建介质区电场的高斯定理方程，其中电场矢量为边

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

update_type：（布尔值）变量更新模式

CreateElementOxidePotentialOnly(device, region, update_type="ault"):

创建介质区电场的高斯定理方程，其中电场矢量为对偶元边

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

update_type：（布尔值）变量更新模式

CreateOxidePotentialContact(device, contact, element_contact=False, attached_to=None, offset_contact=None, is_circuit=False)

创建介质区电场在电极处的边界条件

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

element_contact：（布尔值）电场矢量是否使用对偶元边，默认“否”，即使用边

attached_to：（字符串）是否于其它电极链接，默认“无”

offset_contact：（字符串）电势是与其它电极一同电势错位

is_circuit：（布尔值）是否接入电路

CreateSiliconOxideInterface(device, interface, InterfaceType="continuous"):

创建半导体区域与电介质区域界面模型

参数含义：

device：（字符串）器件名称

interface：（字符串）界面名称

InterfaceType：（布尔值）电场高斯定理界面匹配模式，默认为连续

CreateSiliconSiliconInterface(device, interface):

以连续模式创建电子、空穴连续性方程，以及电场高斯定理

CheckNodeValues(device, region, NodeChecklist=None, Length=5)

打印节点变量以及模型的值

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

NodeChecklist：（列表）变量以及模型名称

Length：（数值）每个变量以及模型打印个数

CheckElementValues(device, region, ElementChecklist=None, Length=5):

打印对偶元边的值

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ElementChecklist：（列表）对偶元边名称

Length：（数值）每个变量以及模型打印个数

CheckEdgeValues(device, region, EdgeChecklist=None, Length=5):

打印边的值

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

EdgeChecklist：（列表）边名称

Length：（数值）每个变量以及模型打印个数

CreateAbsEfield(device, region):

创建电场绝对值，进创建对偶元边“ElectricField _mag”，适用于作图

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

CreateAbsCurrentAnield(device, region, magnitude=False):

创建载流子流速以及电场绝对值

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ElectricFieldMagnitude(device, region):

创建电场绝对值，创建对偶元边“ElectricField _mag”及其对电势的导数，适用于迁移率模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

MicorSaturationMobilityParameters(device, region, V_n_Sat=1.0e11, V_p_Sat=1.0e11, LorentzEField=1.0e-2):

创建或更改用于微电导迁移率模型使用的参数

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

V_n_Sat：（数值）电子饱和迁移率，默认 1.0e11

V_p_Sat：（数值）空穴饱和迁移率，默认 1.0e11

LorentzEField：（数值）洛伦兹饱和场。默认=1.0e-2

CentiSaturationMobilityParameters(device, region, V_n_Sat=1.0e7, V_p_Sat=1.0e7, LorentzEField=1.0e4):

创建微电导迁移率模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

V_n_Sat：（数值）电子饱和迁移率，默认 1.0e7

V_p_Sat：（数值）空穴饱和迁移率，默认 1.0e7

LorentzEField：（数值）洛伦兹饱和场。默认=1.0e4

CreateNormalElectricFieldFromCurrentFlow(device, region, low_curr):

创建电场沿指定载流子电流的分量

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

low_curr：（字符串）指定载流子电流

ProjectSaturationMobility(device, region, ChargeType):

创建饱和迁移率模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ChargeType：（字符串）载流子类型，可选"Electrons"或"Holes"

QSSaturationMobility(device, region, ChargeType, EfieldModel="ElectricField_x"):

创建单方向饱和迁移率模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ChargeType：（字符串），可选"Electrons"或"Holes"

EfieldModel：（字符串）选定的电场方向，可选"ElectricField_x"或"ElectricField_y"

LorentzSaturationMobility(device, region, mu_sat, LorentzEField, ChargeType):

创建洛伦兹饱和迁移率模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

mu_sat：（数值）饱和迁移率大小

LorentzEField：（字符串）选定的电场方向，可选"ElectricField_x"或"ElectricField_y"

ChargeType：（字符串），可选"Electrons"或"Holes"

PooleFrenkelMobilityParameters(device, region, ChargeType, ActivePotential=0, Beta=0.1, Gama=0)

创建或更改用于 PooleFrenkel 迁移率模型使用的参数

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ChargeType：（字符串），可选"Electrons"或"Holes"

ActivePotential：（数值）激活势垒

Beta：（数值）

Gama：（数值）

PooleFrenkelMobility(device, region, ChargeType):

创建 PooleFrenkel 迁移率模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ChargeType：（字符串），载流子类型，可选"Electrons"或"Holes"

CreateElementGaussianQuasiFermi(device, region, TypeName, CarrierName):

根据载流子浓度创建高斯费米能级分布载流子的费米能级的对偶元边模型，不包含静电势，

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

TypeName：（字符串）能级结构类型，可选"GaussianDLkT", "GaussianHomo"，"GaussianALkT", "GaussianLumo"

CarrierName：（字符串）载流子类型，可选"Electrons"或"Holes"

CreateEdgeGaussianQuasiFermi(device, region, TypeName, CarrierName):

根据载流子浓度创建高斯费米能级分布载流子的费米能级的边模型，不包含静电势，

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

TypeName：（字符串）能级结构类型，可选"GaussianDLkT", "GaussianHomo"，"GaussianALkT", "GaussianLumo"

CarrierName：（字符串）载流子类型，可选"Electrons"或"Holes"

CreateNodeGaussianQuasiFermi(device, region, TypeName, SolutionName=None):

根据载流子浓度创建高斯费米能级分布载流子的费米能级的结点模型，不包含静电势，

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

TypeName：（字符串）能级结构类型，可选"GaussianDLkT", "GaussianHomo"，"GaussianALkT", "GaussianLumo"

CarrierName：（字符串）载流子类型，可选"Electrons"或"Holes"

MacCDTMobility(device, region, CarrierName, MacCDTPre=2e-9, MacCo=1, PooleFrenkel=0.26,TrapName=None, MiniValue=1e2, TailState=1e-8):

创建 MacCDT 迁移率模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

CarrierName：（字符串）载流子类型，可选"Electrons"或"Holes"

ActivePotential：（数值）激活势垒

MacCDTPre：（数值）参数，默认 2e-9

MacCo：（数值）参数，默认 1

PooleFrenkel：（数值）参数，默认 0.26

TrapName：（字符串）陷阱态类型，默认为空

MiniValue：（数值）参数，默认 1e2

TailState：（数值）参数，默认 1e-8

ShurHackMobility(device, region, Carrier, TrapName, LRatio=0.01,Exponent=1):

创建 ShurHack 迁移率模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

Carrier：（字符串）载流子类型，可选"Electrons"或"Holes"

TrapName：（字符串）陷阱态类型，默认为空

LRatio：（数值）参数，默认 0.01

Exponent：（数值）参数，默认 1

ElementElectronCurrent2d(device, region)

创建电子电流对偶元边模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ElementElectronContinuityEquation(device, region):

以电子电流对偶元边模型创建电子连续性方程

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ElementHoleCurrent2d(device, region, OrganicCurrent=False):

创建空穴电流对偶元边模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ElementHoleContinuityEquation(device, region):

创建空穴电流对偶元边模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

CreateElementSiliconDriftDiffusion(device, region):

创建基于对偶元边的载流子漂移扩散模型

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ElementContactElectronContinuityEquation(device, contact):

在电极处创建基于对偶元边的电子界条件

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

ElementContactHoleContinuityEquation(device, contact):

在电极处创建基于对偶元边的空穴边界条件

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

CreateElementCurrentContact(device, contact):

在电极处创建基于对偶元边的电流边界条件

参数含义：

device：（字符串）器件名称

region：（字符串）介质区域名称

CreateCapacitorCircuit(contact, attached_to=None, value=0.0, acreal=0.0, resistance=1e0)

创建电容电路结构关系

参数含义：

contact：（字符串）电容名称

attached_to：（字符串）是否于其它电极链接，默认“无”

value：（数值）电容大小，默认 0.0

acreal：（数值）交流电压的实部，默认 0.0

resistance：（数值）所窜联电阻的大小，默认 1

CreateContactCircuit(contact, attached_to=None, value=0.0, acreal=0.0, resistance=1e0)

创建电容电路结构关系

参数含义：

contact：（字符串）电容名称

attached_to：（字符串）是否于其它电极链接，默认“无”

value：（数值）电容大小，默认 0.0

acreal：（数值）交流电压的实部，默认 0.0

resistance：（数值）所窜联电阻的大小，默认 1

CreateTransientCircuit(contact, resistance=1e0):

给电极串联电阻，用于瞬态分析

参数含义：

contact：（字符串）电容名称

resistance：（数值）所窜联电阻的大小，默认 1

PrintCapacitor(device, contact, frequency=[1.0,]):

打印器件中某电极在特定频率下的电容

参数含义：

device：（字符串）器件名称

contact：（字符串）电容名称

frequency：（列表）需要测试的频率

GetSaveFileName(device,FileName):

由器件中对象"Description"获取保存文件的路径及文件名

参数含义：

device：（字符串）器件名称

FileName：（字符串）预设文件名

QSSaveDevice(device, file, ftype="devsim"):

保存器件模拟结果，会调用 GetSaveFileName 生成路径

参数含义：

device：（字符串）器件名称

file：（字符串）文件名

ftype：（字符串）保存类型，可选 devsim', 'devsim_data', 'floops', 'tecplot', 'vtk'，默认"devsim"

ExportParameters(device, file="DeviceParameters")

以 pathon 文件形式输出器件全部参数，会调用 GetSaveFileName 生成路径

参数含义：

device：（字符串）器件名称

file：（字符串）文件名，默认"DeviceParameters"

GetContactBiasName(contact):

获取电极电势名称

参数含义：

contact：（字符串）电极名称

GetContactNodeModelName(contact):

获取电极节点模型名称

参数含义：

contact：（字符串）电极名称

PrintCurrents(device, contact, ContactList=[]):

获取器件电极的电流

参数含义：

device：（字符串）器件名称

contact：（字符串）电容名称

ContactList：（列表）电极名称列表

SetContactDopingOffset(device, contact):

根据硅半导体的玻尔兹曼模型，以净参杂浓度设置电极处欧姆接触的电势初始条件

参数含义：

device：（字符串）器件名称

contact：（字符串）电容名称

GetContactDopingOffset(device, contact):

获取净参杂浓度导致电极处欧姆接触的电势偏移量

参数含义：

device：（字符串）器件名称

contact：（字符串）电容名称

SetContactPotentilOffset(device, contact, PotentialOffset):

置电极处欧姆接触的电势初始条件

参数含义：

device：（字符串）器件名称

contact：（字符串）电容名称

PotentialOffset：（数值）电势偏移量

GetContactPotentilOffset(device, contact):

获取电极的电势偏移量

参数含义：

device：（字符串）器件名称

contact：（字符串）电容名称

SpecificValueInitiateRamp(device, Region, Parameter, abs_error=1e30, rel_error=1e-8, iterations=30, ratio=2):

计算参数引入后的平衡态，如果不能平衡，则将参数除以一定倍数后再次计算

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Parameter：（字符串）区域名称

abs_error：（数值）迭代误差绝对值，默认 1e30

rel_error：（数值）迭代误差相对值，默认 1e-8

iterations：（数值）迭代次数限制，默认 30

ratio：（数值）所除倍数，默认 2

SpecificValueApproach(device, Region, Parameter, SetToValue, abs_error=1e30, rel_error=1e-8, iterations=30, step_lag=0.1, min_setp=0.001, method="Line"):

按照一定步长改变特定参数，计算平衡态，如果不能平衡，则将参数除以 2 后再次计算，直至步长小于特定值后放弃

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Parameter：（字符串）区域名称

abs_error：（数值）迭代误差绝对值，默认 1e30

rel_error：（数值）迭代误差相对值，默认 1e-8

iterations：（数值）迭代次数限制，默认 30

ratio：（数值）所除倍数，默认 2

step_lag：（数值）步长，默认 0.1

min_setp：（数值）最小步长，默认 0.001,

method：（数值）接近方式，默认"Line"

InitialSolve(device, orders=5, type="dc", abs_error=1e30, rel_error=1e-8, iterations=30, tdelta=0.0)

计算平衡态，如果不能平衡，则放大相对误差容许值后再次计算，直至相对误差增大 10 的指数倍后放弃

参数含义：

device：（字符串）器件名称

orders：（数值）限定的指数，默认 5

type：（字符串）计算模式，默认直流"dc"

abs_error：（数值）迭代误差绝对值，默认 1e30

rel_error：（数值）迭代误差相对值，默认 1e-8

iterations：（数值）迭代次数限制，默认 30

tdelta：（数值）用于瞬态模拟的时间步长，默认 0.0

NodeModelVolumeIntegraton(device, NodeDic)

计算特定网格节点模型的积分值即其它计算方式

参数含义：

device：（字符串）器件名称

NodeDic：（字典）包含三个值：去域名"region"，节点名"NodeName"，计算方式"SampleType"，可选体积积分"VolSum"、最大值"Max"、最小值"Min"

3) QSExtraCharge.py

创建缺陷态物理关系模型

CreateGaussianDOS(device, region, Type, Width, Center, BulkDensity, Interface=None, Thickness=0.05)

设置高斯分布能级的基础参数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Type：（字符串）能级类型，可选="GaussianHomo" 或 "GaussianLumo"

Width：（数值）能级分布宽度

Center：（数值）能级中心位置

BulkDensity：（数值）总态密度

Interface：（数值）界面名称，默认为空，反之，则创建界面态

Thickness：（数值）界面态厚度，默认 0.05

FixExtraNodeCharge(device, region, Charge_Name):

将节点模型电荷转变为节点变量

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Charge_Name：（字符串）节点模型电荷名

InitDetrappingGaussianCharge(device, region, Charge_Name, Type):

设置高斯分布陷阱态电荷浓度

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Charge_Name：（字符串）节点模型电荷名

Type：（字符串）选定的陷阱态类型

SetGaussianDOSTrapParameters(device, region, TrapName, TotalDensity, Width, Center, Rate0, LocalScale, HDistance, Interface=None, Thickness=0.001):

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

TrapName：（字符串）陷阱态名称

TotalDensity：（数值）总态密度

Width：（数值）能级分布宽度

Center：（数值）能级中心位置

Rate0：（数值）跃迁尝试频率

LocalScale：（数值）定域化尺度

HDistance：（数值）载流子在能级内部的平均跃迁距离

Interface：（数值）界面名称，默认为空，反之，则创建界面态

Thickness：（数值）界面态厚度，默认 0.05

CreateDonorLikeGaussianTrapEquation(device, region, PooleFrenkel=True, LumoRate=False):

设置高斯分布陷阱态公式

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

PooleFrenkel：（布尔值）是否使用 PooleFrenkel 模型，默认 True,

LumoRate：（布尔值）是否计算与 Lumo 轨道的跃迁关系，默认 False

CreateGaussianTrapNode(device, region, TrapName, LumoRate=False, HomoRate=False, PooleFrenkel=True):

设置高斯分布陷阱态节点模型

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

PooleFrenkel：（布尔值）是否使用 PooleFrenkel 模型，默认 True,

LumoRate：（布尔值）是否计算与 Lumo 轨道的跃迁关系，默认 False

HomoRate：（布尔值）是否计算与 Lumo 轨道的跃迁关系，默认 False

CreateDonorLikeGaussianTrapContactEquation(device, region, contact)

设置高斯分布陷阱态电极公式

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

contact：（字符串）电极名称

CreateGaussianDOSDetrapping(device, region, ChargeModel, ChargeName, DetrappingDistance, Interface):

设置高斯分布陷阱态复合模型

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

ChargeModel：（字符串）电荷类型

ChargeName：（字符串）电荷名称

DetrappingDistance：（字符串）

Interface：（字符串）界面名称

ReplaceExtraPotentialNodeCharge(device, region, *Varables):

替换电场高斯公式中的非载流子部分电荷

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Varables：（字符串）多个电荷名称

InterfaceCharge_NodeModel(device, region, Interface, Charge_Type, Thickness):

设置界面电荷节点模型

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Interface：（字符串）界面名称

Charge_Type：（字符串）电荷类型

Thickness：（数值）界面态厚度

Interface_NodeParameter(device, region, Interface, ParameterName, Thickness):

设置界面节点模型参数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Interface：（字符串）界面名称

ParameterName：（字符串）参数名称

Thickness：（数值）界面态厚度

4) QSFerro.py

创建铁电材料关系模型

CentiMeterFerroParameters(device, region,

SaturationPolarization = 7.749e-6, #C/cm^2

RemanentPolarization = 7.130e-6, #C/cm^2

CoerciveField = 5.0e5, #V/cm

eps_Ferro = 12,

FunctionModel = "tanh"):

设置 Miller 模型的铁电介质参数，以厘米为长度基本单位

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

SaturationPolarization：（数值）饱和极化强度，默认 7.749e-6, #C/cm^2

RemanentPolarization：（数值）剩余极化强度，默认 7.130e-6, #C/cm^2

CoerciveField：（数值）矫顽场，默认 5.0e5, #V/cm

eps_Ferro：（数值）介电常数，默认 12

FunctionModel：（字符串），数学函数，默认"tanh"

MicroMeterFerroParameters(device, region,

SaturationPolarization = 7.749e-14, #C/um^2

RemanentPolarization = 7.130e-14, #C/um^2

CoerciveField = 5e1, #V/um

eps_Ferro = 12,

FunctionModel = "tanh"):

设置 Miller 模型的铁电介质参数，以微米为长度基本单位

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

SaturationPolarization：（数值）饱和极化强度，默认 7.749e-14, #C/um^2

RemanentPolarization：（数值）剩余极化强度，默认 7.130e-14, #C/um^2

CoerciveField：（数值）矫顽场，默认 5e1, #V/um

eps_Ferro：（数值）介电常数，默认 12

FunctionModel：（字符串），数学函数，默认"tanh"

WriteFerroParameters(device, region, SaturationPolarization, RemanentPolarization, CoerciveField, eps_Ferro, FunctionModel)

写入 Miller 模型的铁电介质参数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

SaturationPolarization：（数值）饱和极化强度

RemanentPolarization：（数值）剩余极化强度

CoerciveField：（数值）矫顽场

eps_Ferro：（数值）介电常数

FunctionModel：（字符串），数学函数

CreateFerroRegion(device, region, update_type="ault", FunctionModel="tanh"):

设置铁电区域公式

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

update_type：（字符串）变量更新方式，默认"ault",

FunctionModel：（字符串），数学函数，默认"tanh"

CreateFerroContactEquation(device, contact, attached_to=None, is_circuit=False):

设置铁电区域电极条件

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

attached_to：（字符串）电势是否随其它电极变化，默认"ault",

is_circuit：（布尔值），是否接入电路，默认 False

FerroRegionIterate(device, region, ElementChecklist=None):

设置铁电区域迭代模型

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

ElementChecklist：（列表）输出对偶元边清单

GinzburgLandauFerroParameters(device, region,

GLFerro_rho = 0, #

GLFerro_alpha2 = 0*1e4 , # cm/F to um/F

GLFerro_alpha4 = 0*1e20, # cm5/FC2 to um5/FC2

):

设置朗道-金兹堡铁电模型参数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

GLFerro_rho：（数值） = 0, #

GLFerro_alpha2：（数值）参数，默认 0*1e4 um/F

GLFerro_alpha4：（数值）参数，默认 0*1e20 um5/FC2

CreateGinzburgLandauFerroRegion(device, region, update_type="ault"):

设置铁电区域公式

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

update_type：（字符串）变量更新方式，默认"ault",

FunctionModel：（字符串），数学函数，默认"tanh"

PolarizationRamp(device, FerroRegion, abs_error=1e30, rel_error=1e-8, iterations=30):

计算铁电参数初始平衡态

参数含义：

device：（字符串）器件名称

FerroRegion：（字符串）区域名称

abs_error：（数值）迭代误差绝对值，默认 1e30

rel_error：（数值）迭代误差相对值，默认 1e-8

iterations：（数值）迭代次数限制，默认 30

CapacitorPlotSweep(device, region,

SweepContact,

ChargeContact,

CurrentContact=None,

End_bias=1.0,

step_limit=0.1,

min_step=0.001,

rel_error=1e-8,

abs_error=1e30,

iterations=30,

SaveAs=None,

Checklist=None,

frequency=1e0)

电容器电压扫描

参数含义：

device：（字符串）器件名称

region：（字符串）铁电区域名称

SweepContact：（字符串）扫描电极名称

ChargeContact：（字符串）需输出电荷电极名称

CurrentContact：（字符串）需计算电流电极名称，默认 None,

End_bias：（数值）结束电压，默认 1.0,

step_limit：（数值）初始电压步长，默认 0.1,

min_step：（数值）最小电压步长，默认 0.001,

abs_error：（数值）迭代误差绝对值，默认 1e30

rel_error：（数值）迭代误差相对值，默认 1e-8

iterations：（数值）迭代次数限制，默认 30

SaveAs：（字符串）None,

Checklist：（字符串）None,

frequency：（字符串）1e0

5) QSSchottkyContact.py

创建肖特基电极关系模型

gfi(zeta, s)

高斯费米积分

参数含义：

device：（数值）变量 1

region：（字符串）变量 2

difgfi(zeta,s):

高斯费米积分一阶偏导

参数含义：

zeta：（数值）变量 1

s：（数值）变量 2

gfiinv(g,s):

高斯费米积分反函数

参数含义：

g：（数值）变量 1

s：（数值）变量 2

SetSchottkyContactParamaters2(device, region, contact, ThermVelocities_e=1e11, ThermVelocities_h=1e11)

设置肖特基电极参数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

contact：（字符串）电极名称

ThermVelocities_e：（数值）电子热激发速率，默认 1e11,

ThermVelocities_h：（数值）空穴热激发速率，默认 1e11

SetSchottkyContactParamaters(device, region, contact, ThermVelocities_h=1e11, ThermVelocities_e=1e11, Bandtype="Silicon"):

设置肖特基电极参数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

contact：（字符串）电极名称

ThermVelocities_e：（数值）电子热激发速率，默认 1e11,

ThermVelocities_h：（数值）空穴热激发速率，默认 1e11

Bandtype：（字符串）能带结构类型，默认"Silicon"

CreateSchottkyDriftDiffusionAtContact(device, contact, element_contact=False):

设置肖特基电极边界条件

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

element_contact：（布尔值）电流矢量是否为对偶元边，默认 False

CreateSchottkyElementElectronsContactEquation(device, contact):

以对偶元边电流矢量设置电子肖特基电极边界条件

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

CreateSchottkyElementHolesContactEquation(device, contact)

以对偶元边电流矢量设置空穴肖特基电极边界条件

参数含义：

device：（字符串）器件名称

contact：（字符串）电极名称

CreateSchottkyPotentilContact(device, contact, PotentialOffset):

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

contact：（字符串）电极名称

SetSchottkEquiCharge(device, region, contact):

以当前载流子浓度设置肖特基电极平衡态浓度

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

contact：（字符串）电极名称

6) QSPlotSweep.py

执行器件电压扫描，扫描结果使用 matplotlib 库实现实时显示与图形保存

BackGroundPlot():

设置不实时显示，仅保存图形，并设置 pandas 函数库显示字符宽度

ForGroundPlot():

设置不实时显示，仅保存图形

CreatePlotFigures(name, SubFigureNums=1, figsize=(10, 5),

创建 matplotlib 图像画布，以及子图，并返回对象

参数含义：

name：（字符串）画布名称

SubFigureNums：（整数）子图数量

figsize：（元组）画布大小

CreateMainFigure(name, figsize=(10, 8)):

创建 matplotlib 图像画布，并返回对象

参数含义：

name：（字符串）画布名称

figsize：（元组）画布大小

CreateSubPlot(Figure, Row, Column, num, axeslabel=["Voltage","Current"], xlog=False,ylog=False, title= None):

在 Matplotlib 子图中绘制图像，返回曲线对象

参数含义：

Figure：（对象）画布对象

Row：（整数）画布大小

Column：（整数）画布大小

num：（整数）画布大小

axeslabel：（字符串列表）坐标标签，默认["Voltage","Current"]

xlog：（布尔值）横坐标以对数显示，默认 False

ylog：（布尔值）纵坐标以对数显示，默认 False

title：（字符串）子图名称，默认 None

RefreshSubPlot(SubFigure,Curves,lists,Title=None,ylogScale=False):

更新子图

参数含义：

SubFigure：（对象）子图对象

Curves：（对象）曲线对象

lists：（列表）曲线数组

Title：（字符串）子图名称，默认 None

ylogScale：（布尔值）纵坐标以对数显示，默认 False

CreatePlotCurves(Figure, SubFigure, lists=([0.1,0.2],[0.1,0.2]), ylogScale=False,labels=[]):

绘制曲线

参数含义：

Figure：（对象）画布对象

SubFigure：（对象）子图对象

lists：（列表）曲线数据

ylogScale：（布尔值）纵坐标以对数显示，默认 False

labels：（字符串列表）曲线标签

SavePlotFigures(Figure, FileName, device=None, FileTypes=["pdf"]):

保存图像

参数含义：

Figure：（对象）画布对象

FileName：（字符串）文件名称

device：（字符串）器件名称，默认 None

FileTypes：（字符串列表）保存文件类型，默认["pdf"]

RemovePlotFigures(Figure, FileName, device=None, FileTypes=["svg","pdf"],):

删除保存图像

参数含义：

Figure：（对象）画布对象

FileName：（字符串）文件名称，默认 None

device：（字符串）器件名称

FileTypes：（字符串列表）保存文件类型，默认["svg","pdf"]

UpdateExtraNodeCharge(device, Step_size=0.1, NodeChecklist=None, Length=10)

更新高斯定理公式中的节点变量电荷

参数含义：

device：（字符串）器件名称

Step_size：（数值）迭代电压间隔

NodeChecklist：（节点变量列表）更新前后需要显示的节点变量，默认无

Length：（整数）更新前后需要显示的节点变量长度，默认 10

VoltagePlotSweep(device,

SweepModel = ["Current"],

solvetype = "dc",

SweepContact = None,

YlogScale = True,

CurrentContacts = [],

ChargeContacts = [],

End_bias = 1.0,

step_limit = 0.1,

min_step = 0.01,

rel_error = 1e-8,

abs_error = 1e30,

iterations = 30,

FerroRegion = None,

ElementChecklist= None,

NodeChecklist = None,

Lable = '',

LowBodyBias = False,

CapacitorContacts = [],

frequency = [1.0,],

VolumeIntegrateList = [],

DeviceMonitorList = [],

SaveAll = False,

SaveData = True,

SaveFinal = True,

SaveZero = False,

SaveCurrentVariation=0,

SaveRange = [0,0],

PlotResults = True,

ExtraNodeChargeUpdate= True):

以电压线性变化模式，扫描特定器件的指定电极，并输出电流等随电压的变化关系

参数含义：

device：（字符串）器件名称

SweepModel：（字符串列表）扫描输出的器件属性，默认["Current"],还可包含"Capacitor"、"NodeCharge"

solvetype：（字符串）扫描电压模式，默认"dc"

SweepContact：（字符串）扫描电极名称，默认 None

YlogScale：（布尔值）纵坐标以对数显示，默认 True,

CurrentContacts：（字符串列表）输出电流的电极，默认空

ChargeContacts：（字符串列表）输出电荷的电极，默认空

CapacitorContacts：（字符串列表）输出电容的电极，默认空

End_bias：（数值）终止电压，默认 1.0,

step_limit：（数值）电压初始步长，默认 0.1,

min_step：（数值）电压最小步长，默认 0.01,

rel_error：（数值）迭代误差相对值极限看，默认 1e-8

abs_error：（数值）迭代误差绝对值极限看，默认 1e30

iterations：（数值）迭代次数限制，默认 30

FerroRegion：（字符串）铁电区域名称，默认 None

ElementChecklist：（列表）每次迭代需输出的对偶元边列表，默认 None

NodeChecklist：（字符串）每次迭代需输出的节点模型列表，默认 None

Lable：（字符串）输出文件的额外标签，默认空

LowBodyBias：（字符串）如果存在体电极，是否随着扫描电极一起变化，默认 False

frequency：（数值列表）扫描电容时的频率列表，默认[1.0,],

VolumeIntegrateList：（列表）需要进行体积分的模型列表，默认空

DeviceMonitorList：（列表）需要监视的模型列表，默认空

SaveAll：（布尔值）是否保存每一步的器件数据，默认 False

SaveData：（布尔值）是否保存扫描时 SweepModel 指定的数据，默认 True

SaveFinal：（布尔值）是否保存最后一步的器件数据，默认 True

SaveZero：（布尔值）是否保存扫描电压为 0V 时的器件数据，默认 False

SaveCurrentVariation：（数值）保存器件时电流最小变化量 0,

SaveRange：（列表）仅在指定电压范围保存器件文件，默认全部[0,0],

PlotResults：（布尔值）是否实时显示，默认 True,

ExtraNodeChargeUpdate：（布尔）是否更新额外电荷，默认 True

```
PlotTransient(device,
        VoltateContacts = [],
        CurrentContacts = [],
        ChargeContacts  = [],
        CapacitorContacts= [],
        TimeModel      = "Log",
        minT_Step      = 1e-15,
        maxTime        = 1e-5,
        rel_error      = 1e-8,
        abs_error      = 1e30,
        iterations     = 30,
        FerroRegion    = None,
        ElementChecklist= None,
        Lable          = '',
        SaveAll        = False,
        SaveData       = True,
        SaveFinal      = True,
        SaveCurrentVariation=0,
        SaveRange      = [0,0],
        PlotResults    = True,
        NodeChecklist =[]
        )
```

计算瞬态变化，并显示器件变化以及打印和保存结果

参数含义：

device：（字符串）器件名称

VoltateContacts：（字符串列表）需要监视电压的电极，默认 [],

CurrentContacts：（字符串列表）需要监视电流的电极，默认 [],

ChargeContacts：（字符串列表）需要监视电荷的电极，默认 [],

CapacitorContacts：（字符串列表）需要监视电容的电极，默认 [],

TimeModel ：（字符串）时间变化模式，默认 "Log",

minT_Step ：（数字）最小时间间隔，默认 1e-15,

maxTime ：（数字） 1e-5,

rel_error ：（数字）容忍的相对误差，默认 1e-8,

abs_error ：（数字） 容忍的绝对误差，默认 1e30

iterations ：（数字） 容忍的迭代次数，默认 30,

FerroRegion ：（字符串）铁电区域名称，默认 None,

ElementChecklist：（字符串列表）需要监视的对偶元边模型，默认 None,

Lable ：（字符串） 保存文件的标签，默认''

SaveAll ：（布尔值）是否每一步均保存，默认 False,

SaveData ：（布尔值） 是否保存数据，默认 True,

SaveFinal ：（布尔值） 是否保存最后的结果，默认 True

SaveCurrentVariation：（字符串）保存电流变化的最小间隔，默认 0,

SaveRange ：（字符串）保存文件的电压范围 [0,0],

PlotResults ：（字符串） 是否实时显示计算结果，默认 True,

NodeChecklist：（字符串列表）需要监视的节点模型，默认 None []

7) QSdiode_common.py

创建一维和二维二极管模拟电子器件

CreateMesh(device, region, Length, refinescale):

创建一维二极管结构

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Length：（数值）器件长度

refinescale：（数值）最新小间隔

CreateMesh2(device, region):

创建默认一维二极管结构

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Create2DMesh(device, region):

创建默认二维二极管结构

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Create2DGmshMesh(device, region):

以 Gmsh 文件创 gmsh_diode2d.msh 创建二维二极管结构

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

Create3DGmshMesh(device, region):

以 Gmsh 文件创 gmsh_diode3d.msh 创建三维二极管结构

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

SetNetDoping(device, region, DonorDoping=1e18, AccepterDoping=1e18, Length=1e-5):

设置器件基本参数

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

DonorDoping：（数值）施主参杂浓度，默认 1e18,

AccepterDoping：（数值）受主参杂浓度，默认 1e18,

Length：（数值）金属结位置，默认 1e-5

InitialSolution(device, region, circuit_contacts=None):

计算初始浓度

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

circuit_contacts：（字符串列表）接入电路的电极名称

DriftDiffusionInitialSolution(device, region, circuit_contacts=None)

创建电子、空穴连续性方程

参数含义：

device：（字符串）器件名称

region：（字符串）区域名称

circuit_contacts：（字符串列表）接入电路的电极名称

第五章、 应用举例

本章以铁电薄膜晶体管为列，说明本软件的具体使用步骤。

铁电晶体管共涉及到四个文件：

1) mos_BotContact.geo

用 gmsh 创建网格文件的源文件，具体使用方法请参见 Gmsh 使用说明书

2) mos_BotContact-intLay2.msh

创建的网格文件，用于第三个 python 文件导入网格，创建器件结构

3) create_mos.py 创建器件结构 python 文件

```
#导入模块

from devsim import *
sys.path.append('[存放路径]')
sys.path.append('../')
from QSsimple_physics import *

#设置器件名称
device="mos_BotContact"

#导入 Gmsh 文件，并创建器件
create_gmsh_mesh(file='mos_BotContact-intLay2.msh', mesh=device)
#设置器件结构参数
device_width      =  1.0 #um
gate_width        =  0.8 #um
bulk_thickness    =  4e-2 #um
air_thickness     =  4e-1 #um
oxide_thickness   =  2e-1 #um
diffusion_thickness =  5e-3 #um

x_diffusion_decay  =  5e-3 #um
y_diffusion_decay  =  5e-3 #um
refine_spacing    =  1e-2 #um

x_bulk_left =   0.0
x_bulk_right =  x_bulk_left + device_width
x_center =     0.5 * (x_bulk_left + x_bulk_right)
x_gate_left =   x_center - 0.5 * (gate_width)+x_diffusion_decay
x_gate_right =  x_center + 0.5 * (gate_width)-x_diffusion_decay
x_device_left =  x_bulk_left - air_thickness
x_device_right = x_bulk_right + air_thickness
```

```
y_bulk_top =      0.0
y_oxide_top =    y_bulk_top - oxide_thickness
y_oxide_mid    =    0.5 * (y_oxide_top + y_bulk_top)
y_bulk_bottom=    y_bulk_top + bulk_thickness
y_bulk_mid    =    0.5 * (y_bulk_top + y_bulk_bottom)
y_device_bottom=y_bulk_bottom + air_thickness
y_air_top    =    y_bulk_bottom + air_thickness
y_diffusion    =    y_bulk_top + diffusion_thickness

#命名 Gmsh 区域
add_gmsh_region    (mesh=device, gmsh_name="bulk",    region="bulk",
material="Silicon")
add_gmsh_region    (mesh=device, gmsh_name="oxide",   region="oxide",
material="Oxide"  )
add_gmsh_region    (mesh=device, gmsh_name="air",    region="air",  material="Air")
add_gmsh_contact   (mesh=device, gmsh_name="drain_contact",  region="bulk",
name="drain", material="metal")
add_gmsh_contact   (mesh=device, gmsh_name="source_contact", region="bulk",
name="source", material="metal")
add_gmsh_contact   (mesh=device, gmsh_name="body_contact",   region="air",
name="ground", material="metal")
#命名电极
add_gmsh_contact   (mesh=device, gmsh_name="gate_contact",   region="oxide",
name="gate", material="metal")
add_gmsh_contact   (mesh=device, gmsh_name="attached_source_contact",
region="oxide", name="attached_source", material="metal")
add_gmsh_contact   (mesh=device, gmsh_name="attached_drain_contact",
region="oxide", name="attached_drain",  material="metal")
#命名界面

add_gmsh_interface (mesh=device, gmsh_name="bulk_air_interface",
region0="bulk", region1="air", name="bulk_air")
add_gmsh_interface (mesh=device, gmsh_name="bulk_oxide_interface",
region0="bulk", region1="oxide", name="bulk_oxide")

#完成 Gmesh 文件导入
finalize_mesh(mesh=device)

#创建器件
create_device(mesh=device, device=device)

#参杂浓度参数
bulk_doping    =   1e3   #1um^3
```

```
body_doping    =   1e7   #1um^3
#n doping
drain_doping   =   -1e7   #1um^3
source_doping  =   -1e7   #1um^3
gate_doping    =   -1e8   #1um^3
#设置器件参杂
SetDopingParameters(device,  "bulk",
                        bulk_doping  =bulk_doping,
                        body_doping  =body_doping,
                        drain_doping =drain_doping,
                        source_doping=source_doping,)

#设置器件结构参数
mydict = {}
mydict["drain_doping"] = drain_doping
mydict["body_doping"] = body_doping
mydict["gate_doping"] = gate_doping
mydict["source_doping"] = source_doping
mydict["bulk_doping"] = bulk_doping
mydict["x_gate_left"] = x_gate_left
mydict["x_gate_right"] = x_gate_right
mydict["x_diffusion_decay"] = x_diffusion_decay
mydict["y_diffusion"] = y_diffusion
mydict["y_bulk_bottom"] = y_bulk_bottom
mydict["y_diffusion_decay"] = y_diffusion_decay

#设置参杂浓度模型

node_model(name="DrainDoping",  device=device, region="bulk",
equation="0.25*%(drain_doping)1.15e*erfc((x-%(x_gate_left)1.15e)/%(x_diffusion_de
cay)1.15e)*erfc(-(%(y_diffusion)1.15e-y)/%(y_diffusion_decay)1.15e)" % mydict)
node_model(name="SourceDoping", device=device, region="bulk",
equation="0.25*%(source_doping)1.15e*erfc(-
(x-%(x_gate_right)1.15e)/%(x_diffusion_decay)1.15e)*erfc(-(%(y_diffusion)1.15e-
y)/%(y_diffusion_decay)1.15e)" % mydict)
node_model(name="NetDoping",    device=device, region="bulk",
equation="DrainDoping + SourceDoping  +%(bulk_doping)1.15e + 1" % mydict)
```

4) ferro_BottomContact.py 设置器件参数，以及模拟器件电压扫描文件

```
#导入必备模块
```

```
from devsim import *
import sys, devsim
import os,  tarfile
sys.path.append('[存放路径]')
sys.path.append('../')
from QSsimple_physics import *
from QSPlotSweep import *
from QSFerro import *
from QSramp import *


#设置器件以 128 位扩展精度运行
SetExtendedPrecision()
SetMicroMeterBasicParameters()
#运行创建器件 python 文件

import create_mos
#器件结构名称
device         = "mos_BotContact"
silicon_regions = ("bulk",)
oxide_region    = "oxide"
air_region      = "air"
oxide_regions   = ("oxide",)
regions         = ("bulk", "oxide", air_region)
interfaces      = ("bulk_oxide", "bulk_air")
FerroRegion     = oxide_region
#创建电势以及电场变量
for i in regions:
  CreatePotentialAndFlux(device, i)

#设置硅区域物理参数以及模型
mu=1e5
for i in silicon_regions:
  SetMicroMeterSiliconParameters(device, i, 300, pMobility=mu, nMobility=mu)
  CreateSiliconPotentialOnly(device, i)


#设置氧化物区域物理参数以及模型
for (c,r,k) in (("ground",air_region,1.0),
              ("gate",oxide_region,3.9)):
  SetOxideParameters(device, r, rPermittivity=k)
  CreateOxidePotentialOnly(device, r, "log_damp")
  CreateOxidePotentialContact(device, c)
```

```
#设置硅区域电极物理参数以及模型
contacts=("source", "drain" )
for c in contacts:
  tmp = get_region_list(device=device, contact=c)
  r = tmp[0]
  print("%s %s" % (r, c))
  SetContactDopingOffset(device, c)
  CreateSiliconPotentialOnlyContact(device, c)


#设置氧化物区域电极物理参数以及模型
attached_contacts=(("attached_source","source"),  ("attached_drain", "drain"))
for (i,j) in attached_contacts:
  SetContactPotentilOffset(device, j, GetContactDopingOffset(device, j))
  CreateOxidePotentialContact(device, i, attached_to=j, offset_contact=j)


#设置界面
for i in interfaces:
  CreateSiliconOxideInterface(device, i)


#计算器件初始值
InitialSolve(device,rel_error=1e-9)


#创建硅区域电子、空穴求解变量，以及连续性方程
for i in silicon_regions:
  CreateElectronAndHoleSolutions(device, i)
  CreateSiliconDriftDiffusion(device, i)


#创建硅区域电子、空穴连续性方程电极边界条件
for c in ("source", "drain"):
  tmp = get_region_list(device=device, contact=c)
  r = tmp[0]
  CreateSiliconDriftDiffusionAtContact(device, c)


#设置饱和迁移率模型
V_Sat=1.0e11
for r in ("bulk",):
  DeleteEdgeModelAndDerivatives(device, r, "HoleCurrent")
  ElectricFieldMagnitude(device, r)
```

```
  MicorSaturationMobilityParameters(device, r,V_n_Sat=V_Sat, V_p_Sat=V_Sat)
  QSSaturationMobility(device, r, "Hole")
  # LorentzSaturationMobility(device, r, "mu_n_Sat", "mu_n_eff",  "LorentzEField")
  ElementHoleCurrent2d(device, r)
  ElementHoleContinuityEquation(device, r)


#以对偶元边模型设置电子、空穴连续性方程电极边界条件
for contact in ( "drain", "source"):
  ElementContactHoleContinuityEquation(device, contact)

#计算模型更改后的初始值

InitialSolve(device, rel_error=1e-9)

#设置铁电区域模型

if FerroRegion!=None:
  print("*******Creat Ferro and Solve")
  MicroMeterFerroParameters(device, FerroRegion)
  CreateFerroRegion(device, FerroRegion,  update_type="log_damp")
  CreateOxidePotentialContact(device,"gate", element_contact=True)
  for (i,j) in attached_contacts:
    SetContactPotentilOffset(device, i, GetContactDopingOffset(device, j))
    CreateOxidePotentialContact(device, i, attached_to=j, offset_contact=j,
element_contact=True)

#逐步调整铁电区饱和极化强度，以计算初始值
  SpecificValueInitiateRamp(device, FerroRegion, "SaturationPolarization",
abs_error=1e30, rel_error=1e-9, iterations=30)

#设置电场矢量绝对值模型
for r in regions:
  CreateAbsCurrentAndEfield(device, r, magnitude=True)


#设置需要监视的模型
ElementChecklist=()#("SweepDirectionY","CoerciveSignY","OldPCoefficientY","NewPC
oefficientY",
"NumeratorTanhY","DenominatorTanhY","PolarizationY","PreElectricField_y","Electric
Field_y")
#设置扫描电压范围
Coersive_bias  = 10.00
Other_bias     = -5
#Star_bias      = max(Other_bias,0)+3*Coersive_bias
```

```
Star_bias     = min(Other_bias,0)-3*Coersive_bias
End_bias      = 16 #ther_bias,0)-3*Coersive_bias
#设置器件标签，用于文件夹名称

Description="1e-16V_Sat%0.2e_mu%0.2e_D%sG%s~%s"%(V_Sat, mu, Other_bias,
Star_bias, End_bias )
set_parameter(device=device, name="Description", value=Description)


#保存初始态文件

QSSaveDevice(device, file="InitialSolve.tec",ftype="tecplot")



#设置电压扫描速度
set_parameter(device=device, name="SweepSpeed", value=10)


#设置需要计算电荷的电极
ChargeContacts=["attached_source","attached_drain","gate"]


#设置扫描步骤信息

SweepSettings=(   ("drain", Other_bias, "pre",  0.5, False, False),
              ("gate",  Star_bias,  "star", 0.5, False, True ),
              ("gate",  End_bias,   "forw", 0.1, True,  True ),
              ("gate",  Star_bias,  "back", 0.1, True,  True ),
           )


#开始扫描电压
for (SweepContact, Bias, Lable, step_limit, SaveZero, SaveFinal) in SweepSettings:
  Msg=VoltagePlotSweep(device, SweepContact=SweepContact, CurrentContacts =
["drain"],End_bias=Bias, FerroRegion=FerroRegion, ChargeContacts=ChargeContacts,
iterations=30, step_limit=step_limit, min_step = 0.005, rel_error=1e-18, Lable=Lable,
ElementChecklist=ElementChecklist, SaveAll=False, PlotResults = True)
```