```
###*****###
# Continuity Equation Procedure #


#Solutions Used
#DevPsi - the device potential and is the controlling variable for this pde
#Elec - the electron concentration
#Hole - the hole concentration
#Donor - the ionized donors - can be complicated with partial ionization
#Acceptor - the ionized acceptors - can be complicated with partial ionization

#Solutions Created
#Econd - the conduction band edge
#Eval - the valence band edge

#Parameters Set
#DampValue is set to a thermal voltage at toom temperature
#AbsError is set to 1mV
#RelError is set to 0.01

#Parameters Used
#RelEps is the relative permittivity, and should be in the PDB as a double "$Mat DevPsi RelEps"
#Affinity is the electron affinity, and should be in the PDB as a double "$Mat Affinity"
#Eg is the band gap, and should be in the PDB as a double "$Mat Eg"

#the electronic charge and permittivity of free space
set q 1.60218e-19
set eps0 8.854e-14

# Call this procedure for electron continuity in a material {Mat}
proc ElecContinuity {Mat} {
   global Vt

   pdbSetDouble $Mat Qfn Rel.Error 1.0e-2
   pdbSetDouble $Mat Qfn Abs.Error 1.0e-2
   pdbSetDouble $Mat Qfn DampValue 0.025

   set eqn "ddt(Elec) - ([pdbDelayDouble $Mat Elec mob]) * (Elec+1.0) * grad(Qfn)"

   pdbSetString $Mat Qfn Equation $eqn

   set e "([pdbDelayDouble $Mat Elec Ec])"
   solution add name=Econd solve $Mat const val = ($e)

   set e "([pdbDelayDouble $Mat Elec Nc]) * exp( -(Econd-Qfn) / ($Vt) )"
   solution add name=Elec solve $Mat const val = "($e)"
}

# Call this procedure for hole continuity in a material {Mat}
proc HoleContinuity {Mat} {
   global Vt
```

```
    pdbSetDouble $Mat Qfp Rel.Error 1.0e-2
    pdbSetDouble $Mat Qfp Abs.Error 1.0e-2
    pdbSetDouble $Mat Qfp DampValue 0.025

    set eqn "ddt(Hole) - ([pdbDelayDouble $Mat Hole mob]) * (Hole+1.0) * grad(Qfp)"
    pdbSetString $Mat Qfp Equation $eqn

    set e "([pdbDelayDouble $Mat Hole Ev])"
    solution add name=Eval solve $Mat const val = ($e)

    set e "([pdbDelayDouble $Mat Hole Nv]) * exp( -(Qfp - Eval) / ($Vt) )"
    solution name=Hole solve $Mat const val = "($e)"
}


# Call this procedure when solving a transient simulation with donor traps with energy {Etrap} and
lifetime {tau} and concentration {Ntrap}
proc ElecContinuity_trans {Mat Ntrap Etrap tau} {
    global Vt

    pdbSetDouble $Mat Qfn Rel.Error 1.0e-2
    pdbSetDouble $Mat Qfn Abs.Error 1.0e-2
    pdbSetDouble $Mat Qfn DampValue 0.025

            set kr "(1/$tau)"

            set kf "($kr * 2 / [pdbDelayDouble $Mat Elec Nc] * exp( $Etrap / ($Vt)))"
            #set kr "($kf * [pdbDelayDouble $Mat Elec Nc] / 2 * exp(- $Etrap / ($Vt)))"

    set eqn "ddt(Elec) - ([pdbDelayDouble $Mat Elec mob]) * (Elec+1.0) * grad(Qfn) + ($kf * (Donor * Elec))
- ($kr * ($Ntrap - Donor)) "
    pdbSetString $Mat Qfn Equation $eqn

    set e "([pdbDelayDouble $Mat Elec Ec])"
    solution add name=Econd solve $Mat const val = ($e)

    set e "([pdbDelayDouble $Mat Elec Nc]) * exp( -(Econd-Qfn) / ($Vt) )"
    solution add name=Elec solve $Mat const val = "($e)"

}


# Call this procedure when solving a transient simulation with donor traps with energy {Etrap} and
lifetime {tau} and concentration {Ntrap}
proc DonorContinuity_trans {Mat Ntrap Etrap tau} {
    global Vt

    pdbSetDouble $Mat Donor Rel.Error 1.0e-2
    pdbSetDouble $Mat Donor Abs.Error 1.0e-2
    pdbSetDouble $Mat Donor DampValue 0.025
```

```
        set kr "(1/$tau)"

        set kf "($kr * 2 / [pdbDelayDouble $Mat Elec Nc] * exp( $Etrap / $Vt))"
        #set kr "($kf * [pdbDelayDouble $Mat Elec Nc] / 2 * exp(- $Etrap / ($Vt)))"

  set eqn "ddt(Donor) + ( $kf * (Donor * Elec)) - ($kr * ($Ntrap - Donor))"
  pdbSetString $Mat Donor Equation $eqn

}


solution name=Hole Gas const val=0.0
solution name=Eval Gas const val=0.0
```