Chapter 1

VHDL-AMS IMPLEMENTATION OF A NUMERICAL BALLISTIC CNT MODEL

Dafeng Zhou, Tom J Kazmierski and Bashir M Al-Hashimi
School of Electronics and Computer Science, University of Southampton
Southampton, SO17 1BJ, UK
dz05r,tjk,bmah@ecs.soton.ac.uk

Abstract This contribution presents a VHDL-AMS implementation of a novel numerical carbon nanotube transistor (CNT) modelling approach which relies on a flexible and efficient cubic spline non-linear approximation of the non-equilibrium mobile charge density. The underlying algorithm creates a rapid and accurate solution of the numerical relationship between the charge density and the self-consistent voltage. This leads to a speed-up in the calculation of the current through the channel by about two orders of magnitude without losing much accuracy. The numerical approximation is accurate within less than 1.5% of the normalised RMS error compared with a previously reported theoretical modelling approach. The proposed VHDL-AMS implementation has been used in simulations of a logic inverter in SystemVision to demonstrate the feasibility of applying the spline-based technique in development of efficient and accurate CNT models for applications in circuit-level simulators.

Keywords: VHDL-AMS, ballistic transport, CNT model, circuit level simulation

Introduction

Transistors using carbon nanotubes are expected to become the basis of next generation integrated circuits [1, 2]. These expectations are motivated by the growing difficulties in overcoming physical limits of silicon-based transistors fabricated using current technologies. A number of theoretical models have been created to describe the interplay between different physical effects within the nanotube channel and their effect on the performance of the device [3–8]. The standard methodology of modelling carbon nanotube transistors (CNTs) is to derive the

channel current from the non-equilibrium mobile charge injected into the channel when voltages are applied to the transistor terminals [1]. However, a common problem these models are facing is the complexity of calculating the Fermi-Dirac integral and non-linear algebraic equations which express the relationships between charge densities and the current. Moreover, the channel current between the source and drain is affected not only by the non-equilibrium mobile charge in the nanotube but also by the charges present at terminal capacitances thus adding to the complexity of the current calculation which is a time-consuming iterative approaches. Recently, the standard theoretical methodology has been improved by approaches where the slow Newton-Raphson iterations and the numerical evaluation of the Fermi-Dirac integral are replaced by numerical approximations while still maintaining good performance compared with theories. These new techniques suggest piece-wise approximation of charge densities, either linear [6] or non-linear [9] to simplify the numerical calculation. However, while both these approaches accelerate current calculations significantly, they are not flexible enough to allow the user to control the trade-offs between the modelling accuracy and implementation speed. Here we generalise our earlier piece-wise non-linear approach [9] and propose a cubic spline piece-wise approximation of the non-equilibrium mobile charge density and develop a very accurate technique where a an accuracy better than 1.5% in terms of average RMS error can be achieved with just a 5-piece spline, which compares favourably with the 5% obtained by the simple non-linear approximation [9]. The spline-based approach still achieves a speed up of around two orders of magnitude compared with a reported implementation of the theoretical model [10] and allows an easy trade-off between accuracy and speed. The spline approximation is not only capable of describing performance of ideal ballistic CNT models, but also extendable with non-ballistic effects. The model has been implemented and tested in MATLAB and VHDL-AMS. As an example, we show how our VHDL-AMS model can be used to simulate a CMOS-like inverter made of two complementary CNTs. This illustrates the feasibility of using this novel model in circuit-level simulators for future logic circuit analysis.

1. Mobile charge density and self-consistent voltage

When an electric field is applied between the drain and the source of a CNT, a non-equilibrium mobile charge is generated in the carbon nanotube channel. It can be described as follows[1, 11, 12]:

$$\Delta Q = q(N_S + N_D - N_0) \tag{1.1}$$

where N_S is the density positive velocity states filled by the source, N_D is the density of negative velocity states filled by the drain and N_0 is the equilibrium electron density. These densities are determined by the Fermi-Dirac probability distribution:

$$N_S = \frac{1}{2}\int_{-\infty}^{+\infty} D(E)f(E - U_{SF})dE \tag{1.2}$$

$$N_D = \frac{1}{2}\int_{-\infty}^{+\infty} D(E)f(E - U_{DF})dE \tag{1.3}$$

$$N_0 = \int_{-\infty}^{+\infty} D(E)f(E - E_F)dE \tag{1.4}$$

where $D(E)$ is the density of states, f is the Fermi probability distribution, E represents the energy levels per nanotube unit length, and U_{SF} and U_{DF} are defined as

$$U_{SF} = E_F - qV_{SC} \tag{1.5}$$

$$U_{DF} = E_F - qV_{SC} - qV_{DS} \tag{1.6}$$

where E_F is the Fermi level, q is the electronic charge and V_{SC} represents the self-consistent voltage [1] whose presence in these equations illustrates that the CNT energy band is affected by the external terminal voltages. The self-consistent voltage V_{SC} is determined by the terminal voltages and charges at terminal capacitances by the following non-linear algebraic equation [1, 6]:

$$V_{SC} = -\frac{Q_t + qN_S(V_{SC}) + qN_D(V_{SC}) + qN_0}{C_\Sigma} \tag{1.7}$$

where Q_t represents the charge stored in terminal capacitances and is defined as

$$Q_t = V_G C_G + V_D C_D + V_S C_S \tag{1.8}$$

where C_G, C_D, C_S are the gate, drain, and source capacitances respectively and the total terminal capacitance C_Σ can be derived by

$$C_\Sigma = C_G + C_D + C_S \tag{1.9}$$

2. Numerical piece-wise approximation of the charge density

The standard approach to the solution of equation (1.7) is to use the Newton-Raphson iterative method and in each iteration evaluate the integrals in equations (1.2) and (1.3) to obtain the state densities N_D and N_S. This approach has been proved effective in CNT transistor modelling [6, 10]. However, the iterative computation and repeated integrations consume immense CPU resources and thus are unsuitable for circuit simulation.

Our earlier work [9] proposed a piece-wise non-linear approximation technique that eliminates the need for these complex calculations. It suggested to calculate the charge densities and self-consistent voltage by dividing the continuous density function into a number of linear and non-linear pieces which together compose a fitting approximation of the original charge density curve. Then the V_{SC} equation (1.7) is simplified to a group of linear, quadratic and cubic equations, which can be solved easily and fast.

Although this approach has been shown to be efficient and accurate [9], its weakness is that it requires an optimal fitting process when deciding on the number of approximation pieces and intervals of the ranges which makes the model inflexible and akward to use. Here we propose to use a cubic spline piece-wise approximation to overcome these difficulties.

For a set of n ($n \geq 3$) discrete points (x_0, y_0), (x_1, y_1), ..., (x_{i+1}, y_{i+1}) $(i = 0, 1, ..., n-2)$, cubic splines can be constructed as follows [13]:

$$y = Ay_i + By_{i+1} + C\ddot{y}_i + D\ddot{y}_{i+1} \tag{1.10}$$

where A,B,C and D are the coefficients for each pieces of the cubic spline. For simple demonstration here, the horizontal interval between every two neighbour points is equal to h, then we have $x_1 - x_0 = x_2 - x_1 = ... = x_{i+1} - x_i = h$. Therefore, the cubic spline coefficients can be expressed as functions of x:

$$A \equiv \frac{x_{i+1} - x}{x_{i+1} - x_i} = \frac{x_{i+1} - x}{h} \tag{1.11}$$

$$B \equiv 1 - A = \frac{x - x_i}{x_{i+1} - x_i} = \frac{x - x_i}{h} \tag{1.12}$$

$$C \equiv \frac{1}{6}(A^3 - A)(x_{i+1} - x_i)^2 \tag{1.13}$$

$$D \equiv \frac{1}{6}(B^3 - B)(x_{i+1} - x_i)^2 \tag{1.14}$$

These equations show that A and B are linearly dependent on x, while C and D are cubic functions of x. To derive the $y(x)$ expression, the second-order derivative of y have to be computed via a tridiagonal matrix:

$$\begin{bmatrix} 1 & 4 & 1 & & \\ & 1 & 4 & 1 & \\ & & ... & & \\ & & 1 & 4 & 1 \end{bmatrix} \begin{bmatrix} \ddot{y}_0 \\ \ddot{y}_1 \\ ... \\ \ddot{y}_{n-1} \end{bmatrix} = \frac{6}{h^2} \begin{bmatrix} y_2 - 2y_1 + y_0 \\ y_3 - 2y_2 + y_1 \\ ... \\ y_{n-1} - 2y_{n-2} + y_{n-3} \end{bmatrix} \tag{1.15}$$

Now that the cubic spline coefficients and the second derivative have been obtained, the function of each spline can be derived with the coefficients a_i, b_i, c_i and d_i calculated by using equations (1.11), (1.12), (1.13), (1.14) and (1.15):

$$y_i = a_i x^3 + b_i x^2 + c_i x + d_i \tag{1.16}$$

The two linear regions that extend the cubic splines on both sides can be described as follows:

$$y = y_n, (x > x_n) \tag{1.17}$$

$$y = a_l x + b_l, (x < x_0) \tag{1.18}$$

where $a_l = \ddot{y}_0 = 3a_0 x_0^2 + 2b_0 x_0 + c_0$ and $bl = y_0 - a_l x_0$. To demonstrate the performance of this approach, we have compared the speed and accuracy of an example model with results of other reported approaches.

3. Performance of numerical approximations

An example model which uses three cubic splines, $n = 4$, and two linear pieces at the ends was compared with the theoretical curves calculated from equations (1.2) and (1.3) correspondingly.

To solve the resulting 3rd order polynomial equations, Cardano's method [14] is applied to determine the appropriate root which represents the correct value of V_{SC}.

According to the ballistic CNT transport theory [1, 10] the drain current caused by the transport of the non-equilibrium charge across the nanotube can be calculated using the Fermi-Dirac statistics as follows:

$$I_{DS} = \frac{2qkT}{\pi\hbar}\left[\mathcal{F}_0(\frac{U_{SF}}{kT}) - \mathcal{F}_0(\frac{U_{DF}}{kT})\right] \tag{1.19}$$

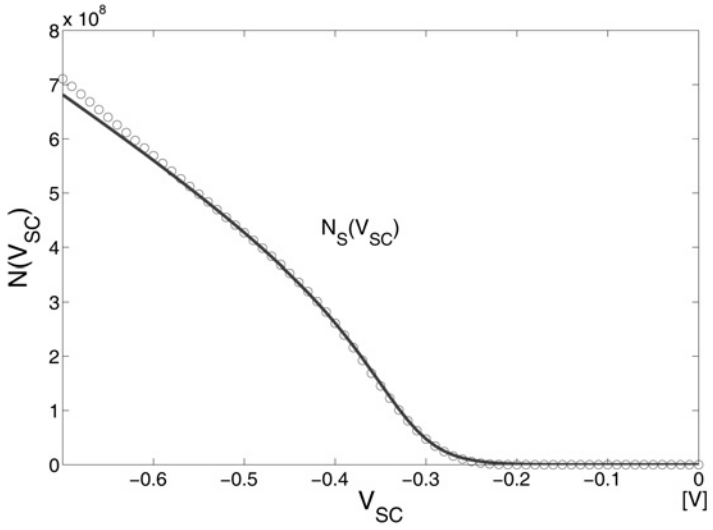

Figure 1.1. Piece-wise cubic spline approximation with $n = 4$ (circlet line) of mobile charge compared with the theoretical result (solid line).

where $\mathcal{F}_0$ represents the Fermi-Dirac integral of order 0, k is Boltzmann's constant, T is the temperature and $\hbar$ is reduced Planck's constant.

Since the self-consistent voltage V_{SC} is directly obtained from the spline model, the evaluation of the drain current poses no numerical difficulty as energy levels U_{SF}, U_{DF} can be found quickly from equations (1.5), (1.6) and I_{DS} can be calculated using:

$$I_{DS} = \frac{2qkT}{\pi\hbar}\left[log(1 + e^{\frac{E_F - qV_{SC}}{kT}}) - log(1 + e^{\frac{E_F - q(V_{SC} - V_{DS})}{kT}})\right] \quad (1.20)$$

These calculations are direct and therefore considerably fast, as there are no Newton-Raphson iterations or integrations of the Fermi-Dirac probability distribution. For performance comparison, we have also tried a 4-piece cubic spline approximation (with $n = 5$) which is expected to be more accurate but slower than the first model. Table 1.1 shows the average CPU times for both models and those from FETToy [10] and previously reported piece-wise models [9], while Table 1.2 compares the accuracy of both numerical model types. It can be seen from Tables 1.1 and 1.2 that although the spline models sacrifice some speed compared with the simple piece-wise non-linear models, they are still more than two orders of magnitude faster than FETToy. They also achieve a much better accuracy than the simple piece-wise non-linear models.

The extent to which the modelling accuracy was compromised by numerical approximation was measured by calculating average RMS errors in the simulations and the results are shown in Table 1.2. As expected, the spline models are more accurate with errors not exceeding 1.0% at $T = 300K$ and $E_F = -0.32eV$ throughout the typical ranges of drain voltages V_{DS} and gate bias V_G. Figure 1.2 shows the I_{DS} characteristics calculated by FETToy compared with the 3-piece spline model.

Table 1.1. Average CPU time comparison between different models.

Loops	*FETToy*	*3-piece PWNL Model*	*4-piece PWNL Model*	*CS Model* $n = 4$	*CS Model* $n = 5$
5	64.43Sec	0.02Sec	0.06Sec	0.57Sec	0.95Sec
10	128.78Sec	0.04Sec	0.12Sec	1.15Sec	1.91Sec
50	642.44Sec	0.19Sec	0.56Sec	5.82Sec	9.59Sec
100	1287.45Sec	0.38Sec	1.12Sec	11.69Sec	19.33Sec

The performance of this approach can be affected by the values of E_F, T, d and terminal voltages. The choice of the number of cubic spline approximation pieces is an obvious trade-off between speed and accuracy as slightly more operations need to be performed with more pieces while the shape of the mobile charge curve is reflected more accurately.

4. VHDL-AMS implementation

The proposed approach has been used to implement both n-type-like and p-type-like CNT transistor models in VHDL-AMS and to simulate a CMOS-like inverter shown in Figure 1.4. The bulk voltage was also considered to take into account the effects on the charge densities generated

Table 1.2. Average RMS errors in piece-wise and cubic spline approximations for $1nm$ nanotube at $E_F = -0.32eV$ and $T = 300K$.

$V_G[V]$	*3-piece PWNL Model*	*4-piece PWNL Model*	*CS Model* $n = 4$	*CS Model* $n = 5$
0.1	4.4%	2.0%	1.3%	0.9%
0.2	3.6%	1.7%	1.0%	0.8%
0.3	2.7%	1.4%	0.8%	0.6%
0.4	1.9%	1.0%	0.6%	0.5%
0.5	1.6%	1.2%	0.9%	0.7%
0.6	2.2%	1.6%	1.1%	1.0%

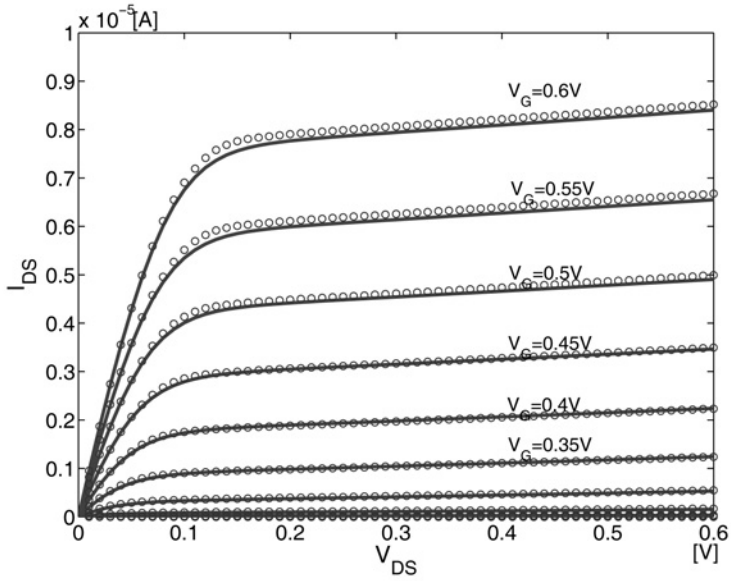

Figure 1.2. Drain current characteristics at $T = 300K$ and $E_F = -0.32eV$ for FETToy(solid lines) and a 3-piece cubic spline approximation (circlet lines).

by the substrate voltage. This is especially important for the p-type-like transistor. Figure 1.3 shows I_{DS} characteristics of the n-type-like transistor implemented in VHDL-AMS which match closely the MATLAB calculations shown in figure 1.2.

The VHDL-AMS testbench for the inverter invokes the two transistors as well as a ramp voltage source and a constant voltage source. The constant source provides the supply voltage V_{CC} for the gate, while the ramp source was used to produce the output characteristic of the inverter. The simulation result is shown in figure 1.5. Considering that the transport characteristics of both transistors are not the same, it is worth noting that the inverter output is not symmetrical at $V_{CC}/2$ due to the stronger n-type-like transistor.

The VHDL-AMS code of the transistor top model is shown below.

```
-- VHDL-AMS model of CNT Transistor I-V Characteristics
-- using cubic spline approximation of S/D charge densities
-- (c) Southampton University 2008
-- Southampton VHDL-AMS Validation Suite
-- Author: Dafeng Zhou, Tom Kazmierski and Bashir M Al-Hashimi
-- School of Electronics and Computer Science
-- University of Southampton
-- Highfield, Southampton SO17 1BJ, United Kingdom
-- Tel. +44 2380 593520   Fax +44 2380 592901
-- e-mail: dz05r@ecs.soton.ac.uk, tjk@ecs.soton.ac.uk
```

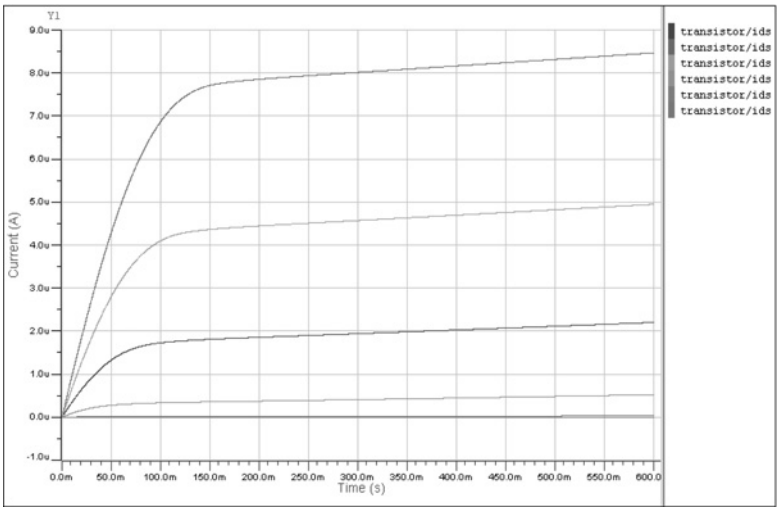

Figure 1.3. VHDL-AMS simulation results on drain current characteristics at $T = 300K$ and $E_F = -0.32eV$ for a 3-piece cubic spline model.

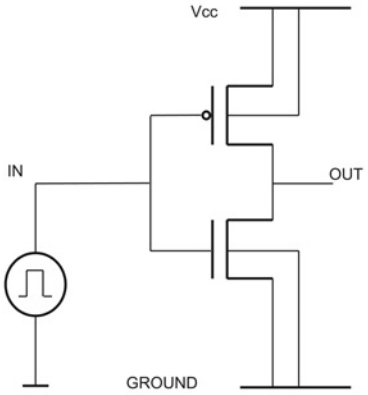

Figure 1.4. Schematics of the simulated inverter.

```
-- Created: 17 October 2007
-- Last revised: November 2008 (by Dafeng Zhou)
--
-------------------------------------------------------------------
--
-- Description:
-- This is a fast numerical model of ballistic transport
-- in carbon nanotube transistors. The default value of the
-- Ef_i parameter (Fermi level) produces n-type-like behaviour;
-- a p-type-like transistor can be obtained by modifying the
-- Fermi level. Package cntcurrent provides the spline data a
-- nd the body of function Fcnt which calculates current Ids
```

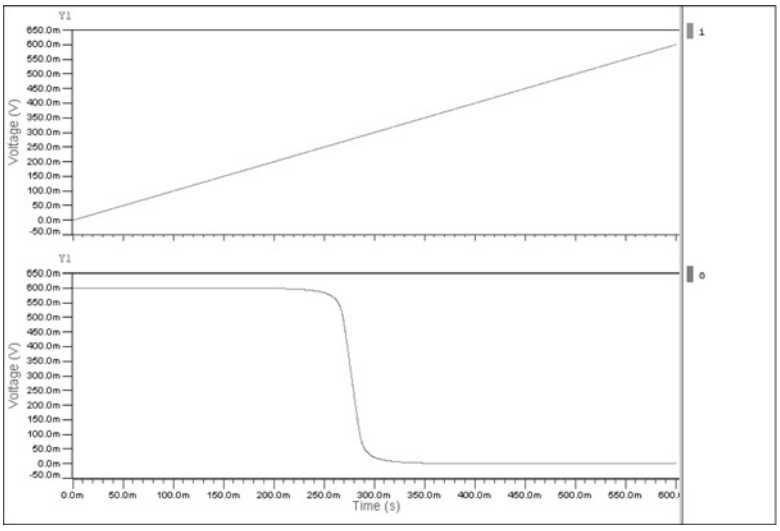

Figure 1.5. Inverter simulation result; input ramps from $0V$ to $0.6V$.

```
-- from the splines.
--
-------------------------------------------------------------------

-- VHDL-AMS Model of Ballistic CNT Transistor

library IEEE;
use IEEE.math_real.all;
use IEEE.electrical_systems.all;

library work;
use work.cntcurrent.all;
use work.SolveVscEquation_pack.all;
use work.coeff_pack.all;


entity CNTTransistor is

    generic( -- model parameters
        T: real := 300.0;
        dcnt: real := 1.0E-9;
        Ef_i: real := -0.32*1.6E-19;
        xmax: real := -0.2;
        xmin: real := -0.5;
        n: integer := 4);
    port (terminal drain, gate, source, bulk: electrical);

end entity CNTTransistor;

architecture Characteristic of CNTTransistor is

    --terminal values
```

```
    quantity Vdi across drain to bulk;
    quantity Vgi across gate to bulk;
    quantity Vsi across source to bulk;
    quantity Ids through drain to source;
begin

    Ids == Fcnt(Vgi, Vsi, Vdi, Ef_i, T, dcnt, xmax, xmin, n);

end architecture Characteristic;
```

The coefficients for cubic spline approximation pieces are derived using a MATLAB script which generates text of the VHDL-AMS package $coeff_pack$. The generated package is included in the simulation.

Combining equations (1.7), (1.16), (1.17) and (1.18), a series of continuous linear and 3rd order polynomial equations to solve the self-consistent voltage are derived using following equations.

$$N_D(V_{SC}) = N_S(V_{SC} - V_{DS}) \tag{1.21}$$

$$\begin{aligned} V_{SC} = \{-Q_t + q(a_i V_{SC}^3 + b_i V_{SC}^2 + c_i V_{SC} + d_i) + q[a_j(V_{SC} - V_{DS})^3 \\ + b_j(V_{SC} - V_{DS})^2 + c_j(V_{SC} - V_{DS}) + d_j] - qN_0\}/C_\Sigma \end{aligned} \tag{1.22}$$

From equation (1.21), $N_D(V_{SC})$ can be treated as an x-axial shift of $N_S(V_{SC})$, and the discrepancy between them is V_{DS}. It can be noticed from equation (1.22) that, when all the parameters are fixed, the value of V_{SC} is determined by only V_{DS} and the spline coefficients. For a given V_{DS}, the summary of $N_S(V_{SC})$ and $N_D(V_{SC})$ can be expressed as $(a_i V_{SC}^3 + b_i V_{SC}^2 + c_i V_{SC} + d_i) + [a_j(V_{SC} - V_{DS})^3 + b_j(V_{SC} - V_{DS})^2 + c_j(V_{SC} - V_{DS}) + d_j]$, which consists of several regions based on the changing of the value of i and j, represented as $QsRange$ and $QdRange$ in an inner function respectively. It can be seen that $QsRange$ and $QsRange$ only change when V_{DS} shifts from one of the spline pieces to another, and in total there are $2n + 1$ regions for the expression. Below are the combination coefficients of different $QsRange$s and $QsRange$s due to the shifting of V_{DS}.

The package listed below contains the code of function $Fcnt$ which solves the spline approximation of the V_{SC} equation (eqn. 1.22) and evaluates the drain current.

```
--Package of Cntcurrent

library IEEE;
use IEEE.math_real.all;
use IEEE.electrical_systems.all;
```

```
library work;
use work.SolveVscEquation_pack.all;
use work.FindQRange_pack.all;
use work.coeff_pack.all;

package cntcurrent is
    function Fcnt(Vgi, Vsi, Vdi, Ef_i, T, dcnt, xmax, xmin: real;
                  n: integer)
    return real;
end package cntcurrent;

package body cntcurrent is
    -- some physical constants:
    constant e0      : real := 8.854E-12;
    constant pi      : real := 3.1415926;
    constant t0      : real := 1.5E-9;
    constant L       : real := 3.0E-8;
    constant q       : real := 1.6E-19;
    constant k       : real := 3.9;
    constant acc     : real := 1.42E-10;
    constant Vcc     : real := 3.0*1.6E-19;
    constant h       : real := 6.625E-34;
    constant hbar    : real := 1.05E-34;
    constant KB      : real := 1.380E-23;

    function Fcnt(Vgi, Vsi, Vdi, Ef_i, T, dcnt, xmax, xmin: real;
                n: integer) return real is
        variable EF, Vd, Vg, Vs, Vds, Ids, Ef_t, Efi, N0, c, Cox,
            Cge, Cse, Cde, Ctot, qC, qCN0, int, Vsc : real;
        variable yy : real_vector(0 to 1);
        variable QsRange, QdRange : integer;

    begin
        Cox := 2.0*pi*k*e0/log((t0+dcnt/2.0)*2.0/dcnt);
        Cge := Cox;
        Cse := 0.097*Cox;
        Cde := 0.040*Cox;
        Ctot:= Cge+Cse+Cde;
        Efi := Ef_i;
        if Efi > 0.0 then
            Ef_t:=-Efi;VD:=-Vdi;VS:=-Vsi;VG:=-Vgi;
        else
            Ef_t:=Efi;VD:=Vdi;VS:=Vsi;VG:=Vgi;
        end if;
        EF:=Ef_t/q;Vd:=Vdi;Vs:=Vsi;Vg:=Vgi;
        N0 := 1.1431E3;
        c := -q*(Vg*Cge+Vs*Cse+Vd*Cde)/Ctot;
        Vds := Vd - Vs;
        qC := q*q/Ctot;
        qCN0 := qC*N0;
        int := (xmax-xmin)/real(n-1);

        -- Derive the ranges of Qs and Qs where the solution of Vsc
            locates
        yy := FindQRange(Vds,q,c,qC,qCN0,xmax,xmin,int,n);
        QsRange := integer(yy(0));
        QdRange := integer(yy(1));
```

```
        -- Calculate the Vsc using Cardone's method from 3rd order
            polynomial and linear equations
        Vsc := SolveVscEquation(Vds,q,c,qC,qCN0,QsRange,QdRange);

        -- Obtain the drain/source current
        if Efi<=0.0 then
            Ids := 4.0*q*KB*T/h*(log(1.0+exp(q*(EF-Vsc)/KB/T))-log
                (1.0+exp(q*(EF-Vsc-Vds)/KB/T)));
        elsif Efi>0.0 then
            Ids := -4.0*q*KB*T/h*(log(1.0+exp(q*(EF-Vsc)/KB/T))-log
                (1.0+exp(q*(EF-Vsc-Vds)/KB/T)));
        else
            Ids := 0.0;
        end if;

        return Ids;
    end function Fcnt;
end package body cntcurrent;
```

5. Conclusion

This contribution proposes to use and investigates the numerical performance of cubic splines in numerical calculations of CNT ballistic transport current with the aim to provide a practical and numerically efficient model for implementation in SPICE-like circuit simulators. The cubic spline approximation is more flexible and easier to use than the earlier piece-wise models [6, 9] and the presented results further reinforce the suggestions that numerical integrations and internal Newton-Raphson iterations can be avoided in the calculation of the self-consistent voltage in the CNT. The cubic spline parameters assure the continuity of the first derivative everywhere and were optimised for fitting accuracy. When compared with FETToy [10], a reference theoretical CNT model, we have demonstrated that the proposed approximation approach, although marginally slower than our earlier models, still leads to a computational cost saving of more than two orders of magnitude while increasing the modelling accuracy. To verify the feasibility of the proposed model, VHDL-AMS implementations for both n-type-like and p-type-like transistors were derived and used to calculate their I_{DS} characteristics as well the output characteristic a simple logic inverter using the SystemVision simulator from Mentor Graphics. The results matched closely those from MATLAB simulations. The new VHDL-AMS model is now available on the Southampton VHDL-AMS Validation Suite website [15] for public use.

Acknowledgments

The authors would like to acknowledge the support of EPSRC/UK for funding this project in part under grant EP/E035965/1.

References

[1] Anisur Rahman, Jing Guo, Supriyo Datta, and Mark S. Lundstrom. Theory of ballistic nanotransistors. *Electron Devices, IEEE*, 50(9):1853–1864, September 2003.

[2] Phaedon Avouris, Joerg Appenzeller, Richard Martel, and Shalom J. Wind. Carbon nanotube electronics. *Proceedings of the IEEE*, 91(11):1772–1784, November 2003.

[3] Arash Hazeghi, Tejas Krishnamohan, and H.-S. Philip Wong. Schottky-Barrier Carbon Nanotube Field-Effect Transistor Modeling. In *Sixth IEEE Conference on Electron Devices*, volume 54, Lausanne, Switzerland, March 2007.

[4] Thao Dang, Lorena Anghel, and Regis Leveugle. Cntfet basics and simulation. In *IEEE Int. conf. on Design and Test of Integrated Systems in Nanoscale Technology (DTIS)*, Tunis, Tunisia, 5-7 September 2006.

[5] Chris Dwyer, Moky Cheung, and Daniel J. Sorin. Semi-empirical SPICE models for carbon nanotube FET logic. In *4th IEEE Conference on Nanotechnology*, Munich, Germany, 16-19 Aug. 2004.

[6] Hamidreza Hashempour and Fabrizio Lombardi. An efficient and symbolic model for charge densities in ballistic carbon nanotube FETs. *IEEE-NANO*, 1:17–20, June 2006.

[7] Bipul C. Paul, Shinobu Fujita, Masaki Okajima, and Thomas Lee. Modeling and analysis of circuit performance of ballistic CNFET. In *2006 Design Automation Conference*, San Francisco, CA, USA, 24-28 July 2006.

[8] Arijit Raychowdhury, Saibal Mukhopadhyay, and Kaushik Roy. A circuit-compatible model of ballistic carbon nanotube field-effect transistors. *Applied Physics Letters*, 23(10):1411–1420, October 2004.

[9] Tom J. Kazmierski, Dafeng Zhou, and Bashir M. Al-Hashimi. Efficient circuit-level modelling of ballistic cnt using piecewise nonlinear approximation of mobile charge density. In *IEEE Int. conf. on Design, Automation and Test in Europe (DATE)*, Munich, Germany, 10-14 March 2008.

[10] Anisur Rahman, Jing Wang, Jing Guo, Sayed Hasan, Yang Liu, Akira Matsudaira, Shaikh S. Ahmed, Supriyo Datta, and Mark Lundstrom. Fettoy 2.0 - on line tool, 14 February 2006. `https://www.nanohub.org/resources/220/`.

[11] Ming-Hsun Yang, Kenneth B. K. Teo, Laurent Gangloff, William I. Milne, David G. Hasko, Yves Robert, and Pierre Legagneux. Advantages of top-gate, high-k dielectric carbon nanotube field-effect transistors. *Applied Physics Letters*, 88(11):113507–1–3, March 2006.

[12] Paul L. McEuen, Michael S. Fuhrer, and Hongkun Park. Single-walled carbon nanotube electronics. *Nanotechnology, IEEE Transactions*, 1(1):78–85, March 2002.

[13] R. Bulirsch and J. Stoer. *Introduction to Numerical Analysis, 2nd edition*. Springer, April 1996.

[14] Ulrich K Deiters. Calculation of densities from cubic equations of state. *AIChE Journal*, 48(4):882–886, April 2002.

[15] Shaolin Wang and Tom J Kazmierski. Southampton VHDL-AMS validation suite, 18 October 2007. `https://www.syssim.ecs.soton.ac.uk/index.htm`.