

Simulación paralela determinista de dispositivos semiconductores 2D basada en esquemas WENO-Boltzmann

J. M. Mantas

R. J. García

M. J. Cáceres

Depto. de Lenguajes y Sistemas Informáticos

Depto. de Matemática Aplicada

E.T.S. Ingeniería Informática y Telecomunicaciones

Facultad de Ciencias

Univ. Granada. 18071 Granada

Univ. de Granada. 18002 Granada

jmmantas@ugr.es

ruben@ugr.es

caceresg@ugr.es

Resumen

Se presenta un simulador paralelo flexible de dispositivos semiconductores 2D que resuelve el sistema de Boltzmann-Poisson para semiconductores mediante un método numérico basado en esquemas en diferencias finitas no oscilatorios de tipo WENO (*Weighted Essentially Non Oscillatory*). Aunque el método ofrece importantes ventajas sobre métodos Monte Carlo, su implementación es muy costosa computacionalmente. Por este motivo se ha abordado su paralelización mediante la descomposición del dominio computacional que concentra la mayor parte de los cálculos. La aplicación paralela que resulta permite simular fácilmente un amplio rango de dispositivos y puede ser usada por ingenieros sin conocimientos sobre el método numérico. Diversos experimentos de simulación de un dispositivo DG-MOSFET sobre un cluster de PCs muestran la eficiencia de la aplicación paralela.

1. Introducción

La simulación de dispositivos semiconductores constituye un área de gran importancia dentro del diseño asistido por ordenador de circuitos electrónicos. Las aproximaciones numéricas a un nivel mesoscópico se basan en la Ecuación de Transporte de Boltzmann (ETB) que describe el flujo de partículas cargadas en un dispositivo semiconductor. La ecuación ETB en

la aproximación semiclásica, tiene la forma:

$$\frac{\partial f}{\partial t} + \frac{1}{\hbar} \nabla_{\mathbf{k}} \varepsilon \cdot \nabla_{\mathbf{x}} f - \frac{\mathbf{e}}{\hbar} \mathbf{E} \cdot \nabla_{\mathbf{k}} f = Q(f), \quad (1)$$

donde f representa la función de densidad de probabilidad (**fdp**) de encontrar electrones en el instante de tiempo t , con coordenada espacial x y vector de onda \mathbf{k} . Las constantes físicas \hbar y \mathbf{e} denotan la constante de Planck dividida por 2π y la carga eléctrica del electrón respectivamente. La función de banda de energía ε viene dada por el modelo de banda no parabólica de Kane, que es una función continua no negativa de \mathbf{k} (véase [4]). $Q(f)$ se denomina el término de colisión ya que modela la interacción de los electrones con el medio.

La electrostática producida por electrones y dopantes se tiene en cuenta mediante el cálculo del campo eléctrico \mathbf{E} de forma autoconsistente usando la ecuación de Poisson:

$$\nabla_{\mathbf{x}} [\epsilon_r(\mathbf{x}) \nabla_{\mathbf{x}} V] = \frac{\mathbf{e}}{\epsilon_0} [\rho(t, \mathbf{x}) - N_D(\mathbf{x})], \quad (2)$$

$$\mathbf{E} = -\nabla_{\mathbf{x}} V, \quad (3)$$

donde $N_D(\mathbf{x})$ es la densidad de dopantes, $\epsilon_r(\mathbf{x})$ denota la función dieléctrica relativa que depende del material (y que puede variar con la posición \mathbf{x}), ϵ_0 es la constante dieléctrica en el vacío, V es el potencial eléctrico y ρ es la densidad de electrones, dada por:

$$\rho(t, \mathbf{x}) = \int_{\mathbb{R}^3} f(t, \mathbf{x}, \mathbf{k}) d\mathbf{k}$$

El sistema de Boltzmann-Poisson para el transporte de electrones en semiconductores viene dado por las ecuaciones (1)-(3).

Las principales herramientas numéricas, usadas por los ingenieros electrónicos para simular (1)-(3) se han basado en métodos de simulación Monte Carlo. Estos métodos no muestran la evolución de la **fdp** hasta el estado estacionario, no aproximan bien regiones del dispositivo casi vacías y producen resultados con ruido numérico. Estos inconvenientes no aparecen en aproximaciones deterministas que resuelven directamente el sistema (1)-(3). No obstante, estos enfoques deterministas deben tratar con la alta dimensionalidad de la ecuación ETB (6 dimensiones más el tiempo), lo cual provoca simulaciones numéricas muy costosas computacionalmente. Para superar este problema, se han propuesto diversos sistemas aproximados y métodos numéricos deterministas (existen referencias en [1]). Sin embargo, hoy en día, existen máquinas paralelas de bajo coste como los clusters de SMPs, que permiten reducir drásticamente los tiempos de simulación. El propósito de este trabajo es mostrar una aplicación paralela basada en esquemas WENO-Boltzmann para realizar de forma precisa un amplio rango de simulaciones de dispositivos 2D. Este trabajo completa el trabajo previo mostrado en [5], de tal forma que la aplicación puede ser usada por cualquier persona interesada en este tipo de simulaciones sin necesidad de poseer conocimientos sobre el esquema numérico.

La sección siguiente describe el esquema numérico en el que se basa el simulador desarrollado. En la sección 3 se muestra cómo se discretiza un dispositivo semiconductor para permitir una implementación eficiente. La sección 4 describe el diseño del simulador paralelo. Los resultados obtenidos al simular un dispositivo DG-MOSFET sobre un cluster de PCs duales se muestran y analizan en la sección 5. Finalmente, la sección 6 presenta las conclusiones y las líneas futuras.

2. Simulador WENO para el sistema Boltzmann-Poisson

La resolución del sistema de Boltzmann-Poisson mostrado en (1)-(3), en el espacio físico 2D, se lleva a cabo mediante la simulación

de un nuevo sistema que se obtiene del sistema previo después de un proceso de adimensionalización seguido de un cambio de variable pseudoesférico para el vector de onda \mathbf{k} , tal como se propuso en [2, 3]. Por supuesto, los resultados se pueden traducir al sistema original deshaciendo los cambios. Tras los cambios mencionados, la nueva función incógnita, Φ se define como:

$$\Phi(t, x, y, \omega, \mu, \phi) = s(\omega)f(t, x, y, \omega, \mu, \phi),$$

donde x e y son las coordenadas espaciales, $\omega \geq 0$ es la energía sin dimensión, $\mu \in [-1, 1]$ es el coseno del ángulo con respecto al eje x y $\phi \in [0, 2\pi]$ el ángulo azimutal, y $s(\omega) = \sqrt{\omega(1 + \alpha_\kappa\omega)(1 + 2\alpha_\kappa\omega)}$. Como resultado de la transformación, la ecuación ETB se puede escribir en forma conservativa para el espacio físico 2D, como sigue:

$$\begin{aligned} \frac{\partial \Phi}{\partial t} + \frac{\partial}{\partial x}(a_1 \Phi) + \frac{\partial}{\partial y}(a_2 \Phi) + \frac{\partial}{\partial \omega}(a_3 \Phi) + \\ \frac{\partial}{\partial \mu}(a_4 \Phi) + \frac{\partial}{\partial \phi}(a_5 \Phi) = s(\omega)C(\Phi), \end{aligned} \quad (4)$$

donde $C(\Phi)$ representa el operador de colisión sin dimensión que viene definido por una ecuación integral sobre términos dependientes de Φ (véase [4]). Los coeficientes a_i , $i = 1, \dots, 5$, que aparecen en los términos de flujo, $\frac{\partial}{\partial \dots}(a_i \Phi)$, son funciones que dependen de t, x, y, ω, μ y ϕ .

Las funciones a_3, a_4 y a_5 dependen de las componentes del vector campo eléctrico, $\mathbf{E} = [E_x, E_y] = [\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y}]$, que se calcula de forma autoconsistente mediante la resolución de la ecuación de Poisson sin dimensión en el dominio espacial 2D:

$$\frac{\partial V}{\partial x^2} + \frac{\partial V}{\partial y^2} = \epsilon(x, y) [n(t, x, y) - n_D(x, y)], \quad (5)$$

donde ϵ es una función que depende de la posición, $n_D(x, y)$ es la densidad de dopantes sin dimensión, V es el potencial eléctrico y $n(t, x, y)$ es la densidad de electrones que viene dada por: $n(t, x, y) = \int_0^\pi \int_0^\infty \int_{-1}^1 \Phi d\mu d\omega d\phi$.

El comportamiento de los dispositivos semiconductores exhibe generalmente regiones con

gradientes fuertemente acusados. Para tratar con esta complejidad en la aproximación de las derivadas, se ha usado un método de resolución muy robusto para leyes de conservación, el esquema de diferencias finitas WENO de quinto orden (WENO5), que se ha empleado ampliamente en diferentes aplicaciones (véase [6, 7]). Este esquema es muy exacto para soluciones suaves pero mantiene al mismo tiempo el carácter no oscilatorio cerca de choques o regiones con gradientes. Por tanto, simulamos numéricamente la Ecuación (4) usando el esquema conservativo WENO5 para aproximar los términos de flujo en cada una de las direcciones.

Con objeto de asegurar la precisión, el dominio computacional 5D (direcciones x, y, ω, μ, ϕ) se discretiza tomando una malla uniforme en cada dirección. Suponiendo que se desea simular el transporte de carga en una zona rectangular con dimensiones $L_x \times L_y$ de un dispositivo semiconductor 2D, y que se va a usar una malla 5D con dimensiones $(N_x + 1) \times (N_y + 1) \times N_\omega \times N_\mu \times N_\phi$ en la resolución numérica, las coordenadas de los puntos de la malla se calculan tal como se muestra en el Cuadro 1, donde $\Delta x = L_x/N_x$, $\Delta y = L_y/N_y$, $\Delta\omega = \omega_{max}/N_\omega$, $\Delta\mu = 2/N_\mu$ y $\Delta\phi = \pi/N_\phi$, donde ω_{max} se escoge en [2] para facilitar las evaluaciones de $C(\Phi)$.

$x_i = i\Delta x,$	$i = 0, 1, \dots, N_x$
$y_j = j\Delta y,$	$j = 0, 1, \dots, N_y$
$\omega_k = (k - 1/2)\Delta\omega,$	$k = 1, \dots, N_\omega$
$\mu_m = (m - 1/2)\Delta\mu - 1,$	$m = 1, \dots, N_\mu$
$\phi_n = (n - 1/2)\Delta\phi,$	$n = 1, \dots, N_\phi$

Cuadro 1: Discretización del dominio 5D.

2.1. Métodos numéricos usados

Para evaluar $C(\Phi)$ y n , en cada instante de tiempo y en cada en cada punto de la malla $(x_i, y_j, \omega_k, \mu_m, \phi_n)$, se usa la fórmula de cuadratura de punto medio, dando lugar a un método conservativo en carga.

Para resolver la ecuación de Poisson (Ecuación 5), se han usado esquemas de diferencias centradas en conjunción con el método iterativo de sobrerrelajación sucesiva (SOR) [9] para

aproximar V y \mathbf{E} . Estos cálculos se han de realizar en todo el dispositivo 2D, y no sólo en los puntos donde hay transporte de carga como ocurre con la Ecuación (4).

La aproximación de los términos de flujo mediante WENO5 es la parte del cómputo que demanda más recursos en todo el algoritmo numérico. En el esquema WENO5, se realiza una aproximación dimensión a dimensión de las derivadas espaciales. Como muestra, supongamos que deseamos aproximar $\frac{\partial}{\partial x}(a_1\Phi)$ en un punto de malla $(x_i, y_j, \omega_k, \mu_m, \phi_n)$. La aproximación de este término de flujo se realiza a lo largo del eje x dejando el resto de variables a valores fijos ($y = y_j, \omega = \omega_k, \mu = \mu_m$ y $\phi = \phi_n$). Sea $g_i = a_1\Phi_i$, $i = -3, \dots, N_x + 3$, donde $\Phi_i = \Phi(t^n, x_i, y_j, \omega_k, \mu_m, \phi_n)$ y a_1 depende de ω_k y μ_m . Cuando $i < 0$ e $i > N_x$, Φ_i viene dado por las condiciones de contorno introducidas en [1]. Entonces se cumple que para $i = 0, \dots, N_x$:

$$\frac{\partial}{\partial x}a_1\Phi_i = \begin{cases} W(g_{i-3}, \dots, g_{i+2}), & \text{si } a_1 > 0 \\ W(g_{i+3}, \dots, g_{i-2}), & \text{si } a_1 \leq 0 \end{cases}$$

donde W es una función no lineal que tiene 6 argumentos. Por tanto, dependiendo del valor de a_1 , se establece un patrón de dependencias ligeramente diferente. La aproximación a $\frac{\partial}{\partial x}a_1\Phi$ en (i, j, k, m, n) depende de Φ_i y de 5 valores vecinos de este valor en la dirección x . Concretamente, si $a_1 > 0$, depende de 3 valores a la izquierda y 2 a la derecha, si $a_1 \leq 0$ depende de 2 valores a la izquierda y 3 a la derecha.

Debido al carácter no oscilatorio del esquema WENO5, un método de Runge-Kutta TVD (*Total Variation Diminishing*) de tercer orden [7], se puede usar de forma conjunta para la discretización en tiempo. Dado el valor de la función en tiempo t^n , Φ^n , este esquema obtiene una aproximación Φ^{n+1} al valor de Φ en $t^{n+1} = t^n + \Delta t$, mediante tres etapas de cálculo, donde las etapas tienen la forma:

$$\Phi^{(s)} = T^{(s)}(\Delta t, \Phi^{(s-1)}, L(\Phi^{(s-1)})), \quad s = 0, 1, 2,$$

donde $\Phi^{n+1} = \Phi^{(2)}$, $T^{(s)}$ es una función lineal que depende de Φ^n (véase [7]) y $L(\Phi) = \frac{\partial\Phi}{\partial t}$ se calcula despejando en la Ecuación (4). Δt se



Figura 1: Esquema de un DG-MOSFET.

ajusta dinámicamente usando una condición (condición CFL) que depende de las funciones a_i y de la malla 5D [1].

3. Discretización del dispositivo

El sistema de Boltzmann-Poisson depende del dispositivo considerado en términos de su geometría y de los materiales. Así por ejemplo, podemos pensar en un Si MOSFET de doble puerta (véase un Si DG-MOSFET 2D en la Figura 1). En este dispositivo hay regiones donde no hay transporte de electrones, (las dos bandas de óxido), por lo que la ecuación ETB sólo se resuelve donde hay transporte (lámina de silicio). En cambio, la ecuación de Poisson se resuelve en todo el dispositivo.

Con objeto de evitar cálculos redundantes, se distinguen dos áreas en el dispositivo 2D. Un área sin transporte de carga, donde sólo se ha de calcular V y \mathbf{E} , y un área con transporte. Como Φ sólo se define en la zona con transporte, el costo computacional del método se concentra en esta zona y la descomposición de los cálculos en este área es fundamental para diseñar la versión paralela.

El dispositivo físico 2D completo (incluyendo la región donde no hay transporte) se discretiza, usando $N_{xall} + 1$ puntos de malla en x y $N_{yall} + 1$ puntos en y . En el área sin transporte, la distancia entre los puntos de malla en x e y no tiene porque ser constante. El espacio físico 2D del área de transporte se discretiza usando un subconjunto de la malla usada para todo el dispositivo con $(N_x + 1) \times (N_y + 1)$ puntos igualmente espaciados en x e y , tal como se definió en el Cuadro 1. Los puntos de inicio y fin en la discretización del espacio físico 2D del área de transporte se asocian a los índices $imin, imax$ en x y $jmin, jmax$ en y , de la dis-

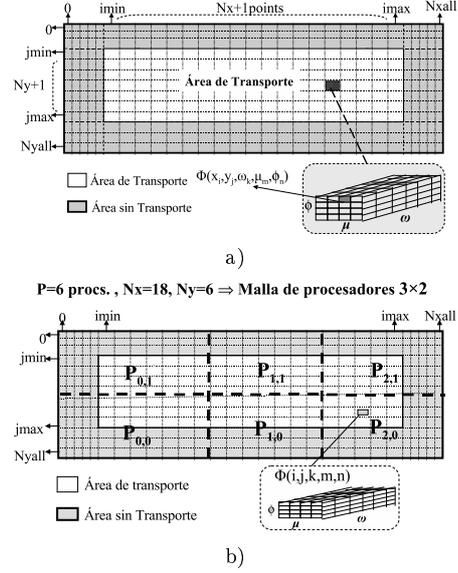


Figura 2: a) Discretización de un dispositivo 2D y b) descomposición del dominio entre los procesadores.

cretización de todo el dispositivo, tal y como se muestra en la Figura 2 a). Para la región de transporte, se ha tomado una malla uniforme en cada una de las direcciones del espacio 5D $(x, y, \omega, \mu, \phi)$ que se definió en el Cuadro 1. Esta elección es necesaria para aplicar el esquema WENO5. Con esta discretización, las principales estructuras de datos usadas son las siguientes:

- Estructuras definidas en todo el dispositivo: Los valores de V , E_x y E_y para cada punto del espacio físico 2D de todo el dispositivo (x_i, y_j) , $i = 0, \dots, imin, \dots, imax, \dots, N_{xall}$, $j = 0, \dots, jmin, \dots, jmax, \dots, N_{yall}$, se representan como matrices 2D con $(N_{xall} + 1) \times (N_{yall} + 1)$ celdas.
- Estructuras exclusivas de la región de transporte:
 - a) Los valores $\Phi(t, x_i, y_j, \omega_k, \mu_m, \phi_n)$ en cada instante de tiempo se almacenan en una matriz con 5 dimensiones $\Phi(i, j, k, m, n)$, con $(N_x + 1) \times (N_y + 1) \times$

$N_\omega \times N_\mu \times N_\phi$ celdas, con los valores de i, j, k, m, n definidos en el Cuadro 1.

b) La densidad de electrones, $n(t, x_i, y_j)$, en cada punto de la malla de transporte y cada instante se modela con una matriz 2D, $n(i, j)$, con $(N_x + 1) \times (N_y + 1)$ celdas.

3.1. Descripción del dispositivo

El usuario de la aplicación describe la información sobre el dispositivo 2D a simular en un archivo, usando una notación fijada, lo que permite aplicar el software desarrollado a cualquier geometría rectangular (o a geometrías expresables mediante regiones rectangulares). Un archivo de descripción se estructura en secciones (GEOMETRY, POTENTIAL, SIMULATION, etc.) y debe incluir:

- **Geometría del dispositivo 2D:** la sección GEOMETRY permite definir las dimensiones del dispositivo completo así como la localización y tamaño de la región de transporte y de los contactos (fuente, drenador y puertas). También se debe incluir en esta sección las dimensiones de la malla de puntos (N_{xall} , N_{yall} , N_x , N_y , N_ω , N_μ , N_ϕ).

- **Condiciones iniciales:** existen secciones específicas para asignar los valores iniciales en cada punto de malla para variables que designan la concentración de dopantes, la constante dieléctrica y el potencial eléctrico. Estos valores se pueden definir asignando valores a regiones rectangulares.

- **Parámetros de la simulación:** como el tiempo de simulación requerido o el número máximo de pasos de tiempo permitidos.

4. Simulador WENO-ETB paralelo

4.1. Estrategia de descomposición

La paralelización de los métodos de resolución se basa en técnicas de descomposición de dominio. Se ha determinado que una buena estrategia consiste en trocear las dimensiones del espacio físico 2D del dispositivo entre los procesadores (descomposición geométrica) usando un esquema de asignación por bloques 2D. Una malla lógica de procesadores 2D se define

automáticamente a partir de los datos de entrada y el número de procesadores disponibles. Las principales estructuras de datos se asignan a dicha malla troceando las dimensiones del espacio físico 2D del dispositivo en bloques rectangulares. No obstante, el equilibrado de la carga se centra en el área de transporte.

Dada una malla 2D de $P_x \times P_y$ procesadores, la matriz $\Phi(i, j, k, m, n)$ se asigna a dicha malla usando bloques rectangulares con $b_x \times b_y \times N_\omega \times N_\mu \times N_\phi$ elementos consecutivos, donde $b_x = \lceil (N_x + 1) / P_x \rceil$ y $b_y = \lceil (N_y + 1) / P_y \rceil$. El mismo tipo de distribución, pero considerando bloques 2D $b_x \times b_y$, se sigue para las matrices 2D definidas en el área de transporte.

Dado que la casi totalidad del costo computacional del esquema se concentra en la región de transporte, la distribución de las matrices 2D con celdas fuera del área de transporte es prácticamente la misma que en el área de transporte, salvo para los procesadores que poseen bloques de puntos situados en las fronteras del dispositivo 2D. Estos procesadores alojan los valores asociados a los mismos puntos de malla que las matrices en el área de transporte, pero también podrían mantener valores de puntos que no pertenecen al área de transporte pero son vecinos al bloque de transporte asignado. La Figura 2 b) muestra cómo se repartiría un dispositivo 2D rectangular, que incluye una región de transporte rectangular mallada con 19×7 puntos, sobre una malla lógica con 3×2 procesadores. En la figura se comprueba como cada procesador gestiona un bloque rectangular con un número similar de puntos en la malla de transporte. Algunos procesadores mantienen además algunos puntos que no pertenecen a la malla de transporte pero estos puntos son vecinos de los puntos de transporte gestionados por el procesador. Con esta opción, la carga computacional está bien repartida entre los nodos de procesamiento ya que la región de transporte requiere casi todo el tiempo de procesamiento. Además, como veremos, los costos de comunicación son relativamente bajos en comparación con el cálculo, y como muchas fases de cálculo no requieren comunicación remota, se puede reutilizar gran parte del código secuencial existente.

4.2. Algoritmo paralelo

A continuación, se muestra un esbozo del algoritmo paralelo que simula la evolución de Φ .

```

Inicializa condiciones iniciales
Para  $(i, j, k, m, n)$  en bloque local
Mientras  $t < t_{final}$ 
  Para  $s = 1, 2, 3$  // Etapas Runge-Kutta
    1. Calcula densidad  $n(i, j)$ 
    2. Calcula  $V(i, j), E_x(i, j), E_y(i, j)$ 
    3.  $L_{i,j,k,m,n} = s(w_k, \mu_m)C(i, j, k, m, n)$ 
    4.  $L_{i,j,k,m,n} - = \sum_{l=3,4,5} \frac{\partial}{\partial v_l} (a_l \Phi)_{i,j,k,m,n}$ 
        // donde  $v_3 = \omega, v_4 = \mu, v_5 = \phi$ 
    5. Intercambia fronteras de  $\Phi$  en  $x$ 
    6.  $L_{i,j,k,m,n} - = \frac{\partial}{\partial x} (a_1 \Phi)_{i,j,k,m,n}$ 
    7. Intercambia fronteras de  $\Phi$  en  $y$ 
    8.  $L_{i,j,k,m,n} - = \frac{\partial}{\partial y} (a_2 \Phi)_{i,j,k,m,n}$ 
    9.  $\Phi(i, j, k, m, n) = T^s(\Phi, \Delta t, L)_{i,j,k,m,n}$ 
    10. Actualiza  $\Delta t; t = t + \Delta t$ 
Fin

```

Algoritmo 1: Alg. paralelo WENO-Boltzmann

Tanto el cálculo de la densidad de electrones $n(i, j)$ (paso 1.) como la evaluación del operador de colisión $C(i, j, k, m, n)$ (paso 3.) requieren evaluar en paralelo la fórmula de cuadratura de punto medio para aproximar las integrales. La distribución de los datos seguida permite que todos los datos necesarios para calcular la integral estén disponibles en el subdominio de un procesador.

Para resolver la ecuación de Poisson (5) (paso 2.), se ha usado una implementación secuencial del método SOR (*Successive OverRelaxation*). En un principio, se usó una versión paralela de este método (red-black SOR [9]) pero, teniendo en cuenta que el resolutor sólo se aplica a dominios 2D de tamaño relativamente pequeño, finalmente se optó por replicar el cálculo de V en todos los procesadores. Esto requiere no sólo replicar el cómputo de V sino que se ha de calcular en paralelo la parte derecha de la Ecuación (5) y replicar el resultado a todos los procesadores. Se ha comprobado que la influencia de esta replicación de datos y cómputo sobre el rendimiento es beneficiosa.

La discretización en tiempo (paso 9.) se lleva a cabo usando una versión del método introdu-

cido en 2.1. Con la distribución seguida para Φ , cada procesador puede actualizar su subdominio de Φ sin comunicarse con otros procesadores en esta fase.

4.3. Cálculo de las derivadas espaciales

Para paralelizar el cálculo de los términos de flujo en cada dimensión (pasos 4.-8.) mediante el esquema WENO5, es necesario considerar las dependencias vistas en el apartado 2.1. Para calcular la derivada con respecto a una dirección en un punto de Φ existía una plantilla asimétrica de 6 puntos (2 puntos a la izquierda y 3 a la derecha o viceversa) que dependía del signo de a_i . Este patrón de dependencias no es tan importante para el cálculo de las derivadas con respecto a ω, μ y ϕ (paso 4.) ya que la distribución de la matriz Φ entre los procesadores asegura que el cálculo de estas derivadas se puede hacer en cada procesador sin comunicación interprocesador.

En cambio, para el cálculo de las derivadas con respecto a x e y (pasos 5.-8.) es necesario intercambiar datos en las fronteras de los subdominios asignados a cada procesador. En general, para evitar la dependencia del signo del coeficiente en las direcciones x e y y reducir el coste de reordenación de datos, se asume una plantilla simétrica de 7 puntos para derivar la solución paralela del método. Por tanto, se asume que el cálculo de la derivada de Φ con respecto a x en un punto (i, j, k, m, n) depende de puntos del conjunto $\{(l, j, k, m, n) : l = i - 3, i - 2, i - 1, i, i + 1, i + 2, i + 3\}$. El mismo patrón de dependencias se usa para aproximar los flujos en y , cambiando el papel de i por j .

Por consiguiente, para calcular las derivadas con respecto a x e y , cada procesador debe comunicar 3 filas o 3 columnas en las fronteras del subdominio que posee de la matriz Φ con cada procesador vecino en la malla de procesadores (véase Figura 3).

5. Resultados experimentales

El simulador se ha programado en C++ aumentado con llamadas a funciones de MPI-1 [10]. Se ha utilizado el compilador gnu de C++ usando opciones recomendadas por AMD pa-

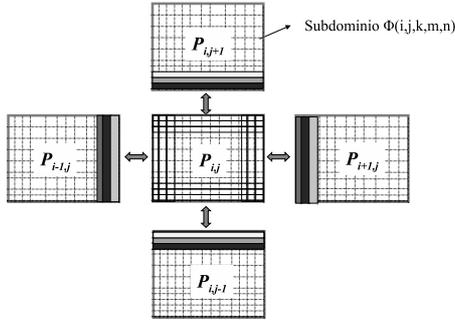


Figura 3: Patrón de comunicación remota para aproximar los flujos en las direcciones x e y .

ra AMD Opteron (-O3 -m64 -march=opteron -ffast-math -funroll-all-loops -fno-vectorize).

El simulador se ha aplicado a un dispositivo DG-MOSFET cuyas dimensiones vienen dadas en la Figura 1. La diferencia de potencial del drenador con respecto a la fuente es de 0.5 voltios y de las puertas con respecto a la fuente de 1 voltio. Se han realizado diversos experimentos numéricos sobre un cluster de 4 AMD Opteron duales, a 2.4 MHz, con 2 GBs. de RAM, conectados con un *switch* Gigabit Ethernet.

Se ha simulado el dispositivo hasta $t = 5$ picosegundos cuando se usa una malla con $N_x = 30$, $N_y = 48$, $N_\omega = 120$, $N_\mu = 12$ y $N_\phi = 12$ para el dominio $(x, y, \omega, \mu, \phi)$ en la región de transporte. Los parámetros que definen el mallado para todo el dispositivo 2D son $N_{xall} = 30$, $N_{yall} = 56$. La Figura 4 muestra cómo se distribuye la concentración de electrones a lo largo de la lámina de silicio del DG-MOSFET en $t = 5$ ps. Se puede observar la existencia de singularidades en las fronteras de ambas puertas.

La Figura 5 a) representa cómo varían los tiempos de ejecución al aumentar el número de procesadores cuando se usan 4 mallas de transporte diferentes. La Figura 5 b) muestra las ganancias en velocidad obtenidas con diversos tamaños de malla y número de procesadores. Los resultados se obtienen al comparar el simulador paralelo con un código monoprocesador altamente optimizado basado en el mis-

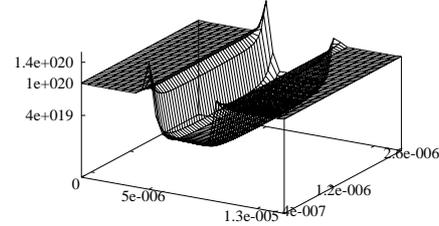


Figura 4: Densidad de electrones en $t = 5$ ps.

mo esquema en un único paso de integración temporal. Los resultados muestran una buena escalabilidad en el rango de procesadores estudiado y una eficiencia paralela entre el 70 % y el 90 % para mallas de interés práctico.

6. Conclusiones y trabajo futuro

Se ha presentado el diseño de un simulador paralelo flexible del sistema Boltzmann-Poisson para dispositivos semiconductores 2D realistas. Los resultados experimentales revelan una buena escalabilidad como resultado de una acertada descomposición geométrica del dominio de datos que equilibra la carga y permite reducir los requisitos de comunicación en las diferentes fases de cálculo. La aplicación resultante permite simular un amplio rango de dispositivos realistas de forma sencilla sin conocer detalles de implementación. El simulador no resulta competitivo en tiempo de simulación con métodos Monte Carlo pero proporciona los resultados de simulación más precisos que se conocen, por lo que puede resultar muy útil para calibrar otros esquemas numéricos.

Se está trabajando en la continuación del trabajo dentro de las siguientes líneas: a) desarrollar un modelo acoplado cuántico determinista que permita simular con precisión dispositivos muy estrechos, en los que la aproximación semiclásica no es válida, b) usar mallas no uniformes en la región de transporte para mejorar prestaciones y precisión mediante el uso de esquemas WENO con interpolación [8] y c) desarrollar una versión de la aplicación para procesadores gráficos programables (GPUs), con objeto de mejorar el rendimiento drásticamente.

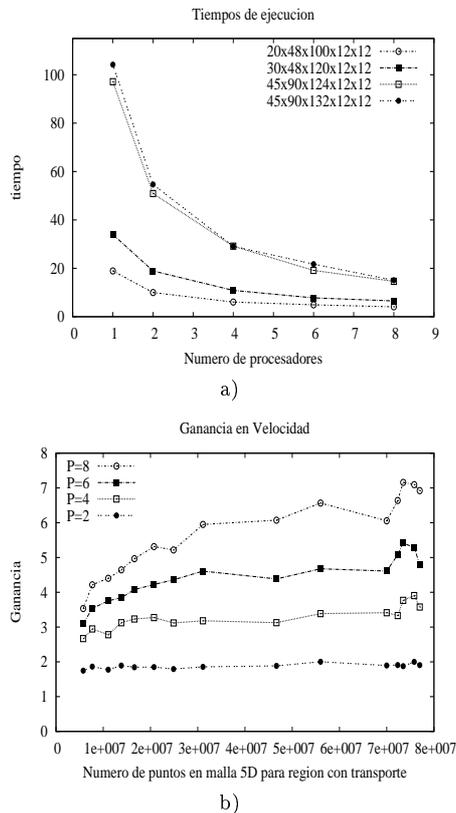


Figura 5: a) Tiempos de ejecución al variar el número de procesadores, b) Ganancias en velocidad al aumentar la densidad del mallado 5D.

Agradecimientos

Los autores J. M. Mantas y R. J. García agradecen el apoyo prestado por el proyecto DGI-MEC TIN2004-07672-C03-02. Los autores M. J. Cáceres y J. M. Mantas agradecen el apoyo prestado por los proyectos DGI-MEC MTM2005-08024 y el proyecto de excelencia de la Junta de Andalucía P06-TIC-01899.

Referencias

[1] Cáceres, M. J., Carrillo, J. A., Gamba, I., Majorana, A., Shu, C.-W. Deterministic kinetic solvers for charged particle transport in semiconductor devices. in Cercig-

nani, C., Gabetta, E. (eds.) Transport Phenomena and Kinetic Theory, Birkhäuser, 151-171.

- [2] Carrillo, J. A., Gamba, I., Majorana, A., Shu, C.-W. A WENO-solver for the transients of Boltzmann-Poisson system for semiconductor devices: performance and comparisons with monte carlo methods. *J. of Comp. Physics*, **184**, 498–525 (2003).
- [3] Carrillo, J. A., Gamba, I., Majorana, A., Shu, C.-W. A direct solver for 2D non-stationary Boltzmann-Poisson systems for semiconductor devices: a mesfet simulation by WENO-Boltzmann schemes. *J. Comp. Electronics*, **2**, 375–380 (2003).
- [4] Carrillo, J.A.; Gamba, I.M.; Majorana, A.; Shu, C.-W.: 2D semiconductor device simulations by WENO-Boltzmann schemes: efficiency, boundary conditions and comparison to Monte Carlo methods, *J. Comput. Phys.*, **214**, (2006), 55–80.
- [5] Mantas, J.M.; Carrillo J.A.; Majorana, A.: Parallelization of WENO-Boltzmann schemes for kinetic descriptions of 2D semiconductor devices. *Mathematics in Industry Springer Series*, **9**, 361-366(2006).
- [6] Shi, J., Zhang, Y.-T., Shu, C.-W., Resolution of high order WENO schemes for complicated flow structures, *J. of Comp. Physics*, **186**, 690–696 (2003).
- [7] Shu, C.-W. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. *L. N. in Mathematics* **1697**, 325-432 (1998)
- [8] Sebastian, K., Shu, C.-W., Multidomain WENO finite difference method with interpolation at subdomain interfaces, *J. Scientific Computing*, **19**, 405-438 (2003).
- [9] Van de Velde, E. F. *Concurrent Scientific Computing*, Springer Verlag, (1994).
- [10] Message Passing Interface Forum, MPI: A Message Passing Interface Standard, Univ. of Tennessee, Knoxville, Tennessee, (1995).