



KHẢO SÁT NGHIỆM CỦA PHƯƠNG TRÌNH POISSON BA CHIỀU TÌM BẰNG THUẬT TOÁN BiCGstab(l)

Lê Thị Ngọc Bảo^{1,2}, Dương Đình Phước¹, Lê Quý Thông², Lê Ngọc Minh², Trần Thiện Lâm^{1*}

¹Trường Đại học Sư phạm, Đại học Huế, 34 Lê Lợi, Huế, Việt Nam

²Trường Đại học Khoa học, Đại học Huế, 77 Nguyễn Huệ, Huế, Việt Nam

Tóm tắt: Bài báo này trình bày việc sử dụng thuật toán BiCGstab(l) để tìm nghiệm của phương trình Poisson ba chiều. Chương trình này sau đó sẽ được tích hợp vào một chương trình mô phỏng linh kiện bằng phương pháp Monte-Carlo tập hợp tự hợp nhằm kiểm tra hiệu năng của nó. Bên cạnh đó, để so sánh tốc độ hội tụ và tính ổn định của thuật toán BiCGstab(l) so với các đặc trưng tương ứng của thuật toán BiCGstab, chúng tôi đã tiến hành khảo sát sự phụ thuộc của chuẩn Euclide của véc-tơ thặng dư vào số vòng lặp của chương trình con Poisson. Kết quả thu được cho thấy chương trình giải phương trình Poisson dựa trên thuật toán BiCGstab(l) cho kết quả mô phỏng chính xác hơn như khi sử dụng thuật toán BiCGstab.

Từ khóa: Mô phỏng linh kiện bán dẫn, phương trình Poisson ba chiều, thuật toán BiCGstab(l), phương pháp Monte-Carlo

1 Giới thiệu

Nền khoa học công nghệ hiện nay đang phát triển rất mạnh mẽ dựa trên nền tảng là các linh kiện bán dẫn na-nô, vì vậy lĩnh vực nghiên cứu các linh kiện loại này đã và đang thu hút sự quan tâm nghiên cứu của nhiều nhà khoa học [1], [2], [3], [4]. Việc nghiên cứu lý thuyết các linh kiện bán dẫn na-nô cho phép dự đoán các tính chất của linh kiện từ đó đưa ra các thiết kế tối ưu để đưa vào sử dụng thực tiễn. Các phương pháp nghiên cứu lý thuyết linh kiện na-nô thường được sử dụng có thể được kể đến như phương pháp Monte-Carlo tập hợp tự hợp, phương pháp các phương trình cân bằng và mô hình kéo theo – khuếch tán [5]. Trong các phương pháp này, Monte-Carlo tập hợp tự hợp có nhiều ưu điểm thể hiện độ chính xác cao và ổn định. Đây là phương pháp bán cổ điển vì các tốc độ tán xạ được tính toán dựa trên quy tắc vàng Fermi và các đại lượng động lực học được xác định từ các phương trình chuyển động của Newton. Ngoài ra, các cơ chế tán xạ được lựa chọn ngẫu nhiên như xảy ra trong thực tế. Việc mô phỏng theo phương pháp này càng chính xác nếu số hạt được mô phỏng càng lớn.

Một phần quan trọng trong việc thực hiện mô phỏng Monte-Carlo tập hợp tự hợp cho linh kiện bán dẫn na-nô đó là giải phương trình Poisson. Phương trình này mang thông tin về sự phân bố điện thế trong linh kiện theo sự phân bố điện tích của linh kiện đó. Để có thể giải

* Liên hệ: tranthanhtamtoan@gmail.com

được, phương trình Poisson thường được làm rời rạc hóa theo phương pháp sai phân hữu hạn để trở thành một hệ phương trình đại số tuyến tính [5]. Đây là một hệ rất lớn các phương trình đại số với hàng chục thậm chí hàng trăm ngàn biến số. Hệ này có thể được giải bằng các phương pháp Jacobi, Gauss – Seidel, SOR, đa lưới (multigrid), iLU [5], hay bằng các phương pháp lặp trong không gian con Krylov như CGS, BiCG, BiCGstab, BiCGstab2, GPBiCG, TFQMR, BiCGstab(3) và BiCGstab(l) [6]. Một số tác giả đã sử dụng phương pháp BiCGstab, CGS, TFQMR và BiCGstab(3) [4] để giải phương trình Poisson và đã thu được kết quả khá chính xác với thời gian tính toán ngắn. Thuật toán BiCGstab(l) là một thuật toán được cải tiến từ thuật toán BiCGstab và được đánh giá là có tốc độ hội tụ về nghiệm nhanh hơn và mang lại độ chính xác cao hơn [7]. Bài báo này trình bày việc nghiên cứu áp dụng thuật toán BiCGstab(l) vào việc giải phương trình Poisson, khảo sát độ chính xác và ổn định của phương pháp này so với các phương pháp đã được sử dụng trước đây như BiCGstab, CGS, TFQMR và BiCGstab(3) [4].

2 Giải phương trình Poisson ba chiều bằng thuật toán BiCGstab(l)

Một giai đoạn trong quá trình mô phỏng là tìm điện thế của các điểm nút trong mô hình linh kiện gây ra bởi điện tích tại các điểm nút đó và điện trường ngoài đặt vào linh kiện. Phương trình tổng quát để tìm điện thế là phương trình Poisson

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2} = -\frac{\rho}{\epsilon_s}, \tag{1}$$

trong đó ρ là mật độ điện tích, ϵ_s là hằng số điện môi tĩnh và φ là điện thế. Thực hiện sai phân hữu hạn, ta chia mô hình linh kiện thành các ô lưới, theo phương Ox có N_x nút lưới, phương Oy và Oz có cùng số nút lưới là N_y . Kích thước các chiều của ô lưới được lựa chọn cho đơn giản là $\Delta x = \Delta y = \Delta z$. Ta thu được phương trình cho một nút lưới như sau

$$\varphi_{i-1,j,k} + \varphi_{i,j-1,k} + \varphi_{i,j,k-1} - 6\varphi_{i,j,k} + \varphi_{i+1,j,k} + \varphi_{i,j,+1k} + \varphi_{i,j,k+1} = -\frac{\rho_{i,j,k}}{\epsilon_s} \Delta x^2, \tag{2}$$

trong đó $i = \overline{1, N_x}, j = \overline{1, N_y}, k = \overline{1, N_z}$. Phương trình (2) có thể được biểu diễn dưới dạng ma trận như sau

$$\mathbf{A}\varphi = \mathbf{B}, \tag{3}$$

trong đó \mathbf{A} là một ma trận vuông ($N \times N$) của các hệ số, $\boldsymbol{\varphi}$ là ma trận ($N \times 1$) của các giá trị điện thế của các điểm nút và \mathbf{B} là ma trận ($N \times 1$) của các hệ số ở vế phải, với $N = N_x N_y N_z$. Trong quá trình thực hiện lập trình giải phương trình (3) theo phương pháp BiCGstab(l) nói riêng và các phương pháp lập trong không gian con Krylov nói chung ta không cần nhập ma trận \mathbf{A} . Các phép tính nhân ma trận \mathbf{A} với véc-tơ $\boldsymbol{\varphi}$ thường xuyên được thực hiện rất đơn giản thông qua việc sử dụng các phương trình (2). Giải thuật BiCGstab(l) được trình bày trong Bảng 1 [7].

Bảng 1. Thuật toán BiCGstab(l) [7]

<p>$k = l,$</p> <p>Chọn $x_0, \tilde{r}_0,$</p> <p>Tính $r_0 = b - Ax_0,$</p> <p>Lấy $\mathbf{u}_{-1} = 0, \mathbf{x}_0 = x_0, \rho_0 = 1, \alpha = 0, \omega = 1$</p> <p>Lặp lại cho đến khi $\ \mathbf{r}_{k+l}\$ đủ nhỏ</p> <p>$k = k + l,$</p> <p>Đặt $\hat{u}_0 = \mathbf{u}_{k-1}, \hat{r}_0 = \mathbf{r}_k$ và $\hat{x}_0 = \mathbf{x}_k.$</p> <p>$\rho_0 = -\omega\rho_0$</p> <p>For $j = 0, \dots, l-1$ (Bi-CG PART)</p> <p>$\rho_1 = (\hat{r}_j, \tilde{r}_0), \beta = \beta_{k+j} = \alpha \frac{\rho_1}{\rho_0}, \rho_0 = \rho_1$</p> <p>For $i = 0, \dots, j$</p> <p>$\hat{u}_i = \hat{r}_i - \beta \hat{u}_i$</p> <p>End</p>	<p>For $j = 1, \dots, l$ (mod.G.S) (MR PART)</p> <p>For $i = 1, \dots, j-1$</p> <p>$\tau_{ij} = \frac{1}{\sigma_i} (\hat{r}_j, \hat{r}_i)$</p> <p>$\hat{r}_j = \hat{r}_j - \tau_{ij} \hat{r}_i$</p> <p>End</p> <p>$\sigma_j = (\hat{r}_j, \hat{r}_j), \gamma'_j = \frac{1}{\sigma_j} (\hat{r}_0, \hat{r}_j)$</p> <p>End</p> <p>$\gamma_i = \gamma'_i, \omega = \gamma_i$</p> <p>For $j = l-1, \dots, 1$ ($\tilde{\gamma} = T^{-1} \tilde{\gamma}'$)</p> <p>$\gamma_j = \gamma'_j - \sum_{i=j+1}^{l-1} \tau_{ij} \gamma_{i+1}$</p> <p>End</p> <p>For $j = 1, \dots, l-1$ ($\tilde{\gamma}'' = T \tilde{\gamma}$)</p> <p>$\gamma_j'' = \gamma_{j+1} + \sum_{i=j+1}^{l-1} \tau_{ij} \gamma_{i+1}$</p>
--	--

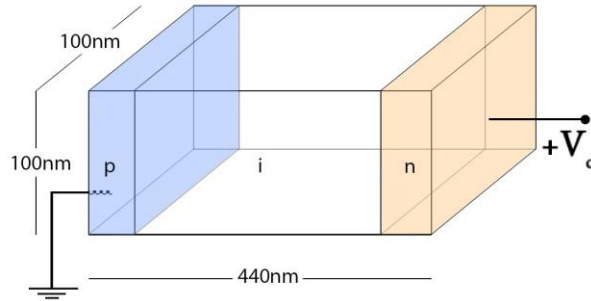
$\hat{u}_{j+1} = A\hat{u}_j$ $\gamma = (\hat{u}_{j+1}, \tilde{r}_0), \alpha = \alpha_{k+j} = \frac{\rho_0}{\gamma}$ <p>For $i = 0, \dots, j$</p> $\hat{r}_i = \hat{r}_i - \alpha \hat{u}_{i+1}$ <p>End</p> $\hat{r}_{j+1} = A\hat{r}_j, \hat{x}_0 = \hat{x}_0 + \alpha \hat{u}_0$ <p>End</p>	<p>End</p> <p>(update)</p> $\hat{x}_0 = \hat{x}_0 + \gamma_1 \hat{r}_0, \hat{r}_0 = \hat{r}_0 - \gamma_1' \hat{r}_1, \hat{u}_0 = \hat{u}_0 - \gamma_1 \hat{u}_1$ <p>For $j = 1, \dots, l-1$</p> $\hat{u}_0 = \hat{u}_0 - \gamma_j \hat{u}_j \text{ (BLAS2 hoặc BLAS3)}$ $\hat{x}_0 = \hat{x}_0 + \gamma_j'' \hat{r}_j \text{ (_GEMV_GEMM)}$ $\hat{r}_0 = \hat{r}_0 - \gamma_j' \hat{r}_j$ <p>End</p> <p>Put $\mathbf{u}_{k+l-1} = \hat{u}_0, \mathbf{r}_{k+l} = \hat{r}_0$ and $\mathbf{x}_{k+l} = \hat{x}_0$</p>
---	--

Chúng tôi đã xây dựng một chương trình con giải phương trình Poisson ba chiều hoàn toàn mới sử dụng thuật toán BiCGstab(l) được trình bày trong Bảng 1. Các yếu tố của ma trận **A** được nhập tự động vì vậy trong quá trình chạy chương trình ta chỉ cần nhập các yếu tố của ma trận **B**. Các bước tính toán được thực hiện tuần tự theo thứ tự trong giải thuật để tìm phân bố điện thế trong mô hình linh kiện, chính là các nghiệm của phương trình Poisson. Để khảo sát độ chính xác và ổn định của chương trình con giải phương trình Poisson ba chiều này, chúng tôi tiến hành tích hợp chương trình con vào một chương trình mô phỏng linh kiện có kích thước nano do chúng tôi đã phát triển trước đây [1] và tiến hành mô phỏng một linh kiện cụ thể. Linh kiện được chọn để khảo sát là đi-ốt p-i-n bán dẫn GaAs, là linh kiện đã được khảo sát chi tiết trước đây để tiện so sánh [1], [2], [3], [4]. Việc phân tích các kết quả mô phỏng sẽ giúp đánh giá chi tiết độ chính xác và độ ổn định của chương trình con.

3 Kết quả mô phỏng và thảo luận

Mô hình linh kiện đi-ốt p-i-n bán dẫn GaAs gồm một lớp bán dẫn thuần (i) nằm giữa hai lớp bán dẫn pha tạp loại p và loại n được minh họa như Hình 1, trong đó mỗi lớp có độ dày lần lượt là d_i, d_p và d_n . Mật độ pha tạp acceptor và donor tương ứng là N_A và N_B . Chương trình mô phỏng sẽ xác định trạng thái cân bằng nhiệt theo thời gian thực rồi mới chiếu xung laser vào

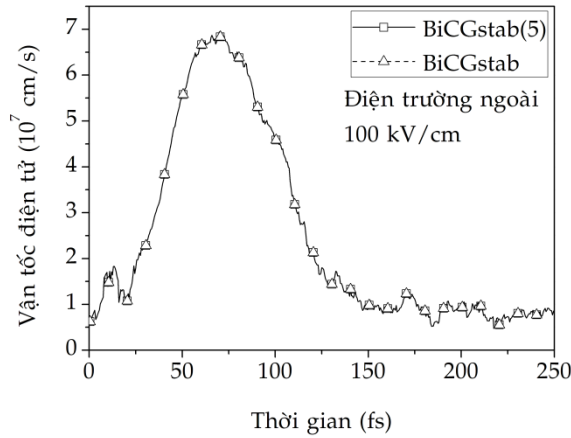
linh kiện để tạo các hạt tải điện là điện tử và lỗ trống tại lớp bán dẫn thuần (i). Các thông số và điều kiện mô phỏng được trình bày trong tài liệu [3].



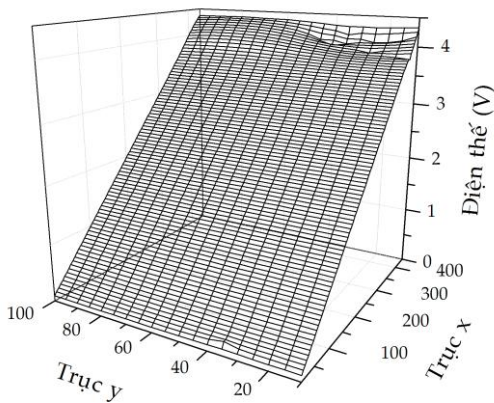
Hình 1. Mô hình đi-ốt p-i-n GaAs [3]

Chúng tôi sẽ lần lượt so sánh các kết quả thu được bằng chương trình mô phỏng tích hợp thuật toán BiCGstab(l) và các chương trình tích hợp các thuật toán khác cho việc tìm nghiệm của phương trình Poisson. Hình 2 cho kết quả so sánh vận tốc của điện tử thu được bằng các chương trình mô phỏng ba chiều sử dụng thuật toán BiCGstab và BiCGstab(l). Kết quả thu được gần như trùng khớp với các kết quả đã được công bố ở các công trình nghiên cứu trước đây [2], [3], [4]. Điều này minh chứng cho tính hiệu quả của phương pháp giải phương trình Poisson bằng phương pháp BiCGstab(l).

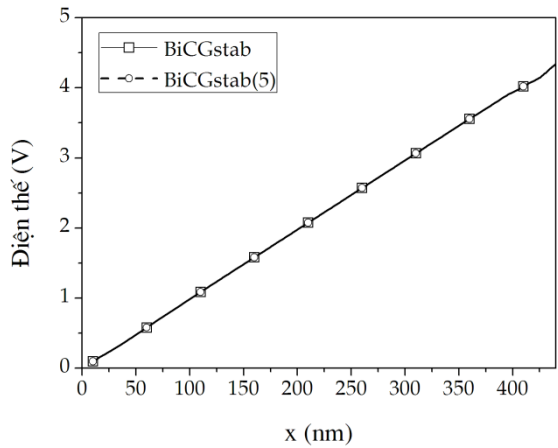
Hình 3 biểu diễn điện thế tại các điểm trên mặt cắt $z = 10$ nm ứng với điện trường ngoài $E_{ex} = 100$ kV/cm. Đồ thị diễn ra hiện tượng biến dạng của lưới điện thế do điện trường ngoài thiết lập gây ra bởi các điện tích phân bố trong linh kiện. Nơi nào có mật độ điện tích càng lớn thì sẽ làm lưới điện thế biến dạng càng nhiều. Chiều biến dạng sẽ phụ thuộc vào dấu của các điện tích tại các nút lưới. Hình 4 biểu diễn phân bố điện thế dọc theo trục Ox , $y = z = 10$ nm ứng với điện trường ngoài $E_{ex} = 100$ kV/cm trong trường hợp chương trình mô phỏng có sử dụng thuật toán BiCGstab(5) và BiCGstab. Kết quả của hình vẽ cho thấy điện thế thu được có giá trị khá gần nhau tại tất cả các điểm lưới của linh kiện trong hai chương trình mô phỏng có sử dụng các thuật toán nói trên.



Hình 2. Vận tốc trôi dạt của điện tử theo phương Ox như là hàm của thời gian với $E_{ex} = 100$ kV/cm theo các chương trình mô phỏng ba chiều sử dụng thuật toán BiCGstab và BiCGstab(l).



Hình 3. Phân bố điện thế không gian trong đi-ốt p-i-n bán dẫn GaAs tại các điểm nút trên mặt cắt $z = 10$ nm ứng với $E_{ex} = 100$ kV/cm.

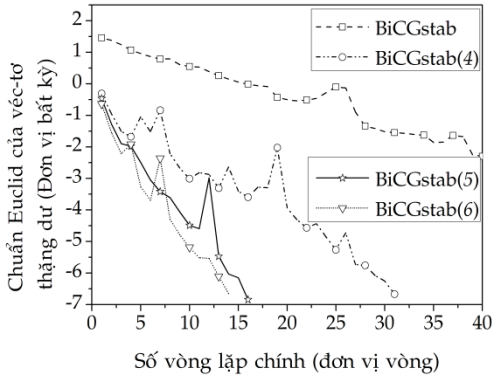


Hình 4. Phân bố điện thế dọc theo trục Ox , $y = z = 10$ nm ứng với điện trường ngoài $E_{ex} = 100$ kV/cm trong trường hợp chương trình mô phỏng có sử dụng thuật toán BiCGstab(5) và BiCGstab.

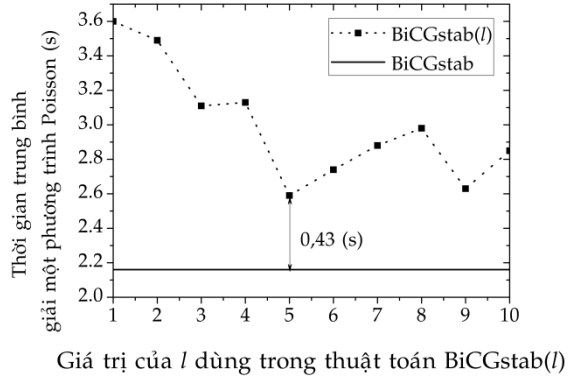
Để so sánh hơn nữa mức độ chính xác của hai phương pháp BiCGstab và BiCGstab(l), chúng tôi khảo sát chuẩn Euclide của véc-tơ thặng dư tương ứng trong hai trường hợp. Chuẩn Euclide của véc-tơ thặng dư là một hàm của số vòng lặp chính của chương trình con Poisson và được tính theo công thức [7]

$$\|r\|_2 = \sqrt{r^T r}, \tag{4}$$

với $r = b - A\phi$ là véc-tơ thặng dư. Từ đồ thị Hình 5 ta thấy rằng thuật toán BiCGstab(l) cần ít vòng lặp chính hơn để thu được nghiệm có cùng một chuẩn Euclide của véc-tơ thặng dư khi so sánh với chương trình con Poisson sử dụng thuật toán BiCGstab.



Hình 5. So sánh chuẩn Euclide của véc-tơ thặng dư trong trường hợp chương trình con Poisson dùng thuật toán BiCGstab và BiCGstab(l)



Hình 6. Thời gian trung bình trên một lần giải phương trình Poisson theo thuật toán BiCGstab và BiCGstab(l) với $l = \overline{1, 10}$

Để so sánh độ chính xác của các giá trị điện thế tại các điểm trên cùng đường thẳng $y = z = 10$ nm tìm được bằng hai chương trình con Poisson dùng thuật toán BiCGstab và BiCGstab(l) chúng tôi vẽ đồ thị biểu diễn độ lớn của các thành phần của véc-tơ thặng dư của tất cả các điểm dọc trục Ox ở Hình 5. Đồ thị biểu diễn độ lớn của các thành phần của véc-tơ thặng dư trong trường hợp chương trình con Poisson dùng thuật toán BiCGstab(l) gần giá trị 0 và đồng đều cho các điểm khác nhau dọc trục Ox . Trong khi đó chương trình con Poisson dùng thuật toán BiCGstab cho kết quả xa giá trị 0 và kém đồng đều hơn. Kết quả này một lần nữa khẳng định rằng chương trình con Poisson sử dụng thuật toán BiCGstab(l) cho ta kết quả mô phỏng chính xác hơn khi dùng giải thuật BiCGstab.

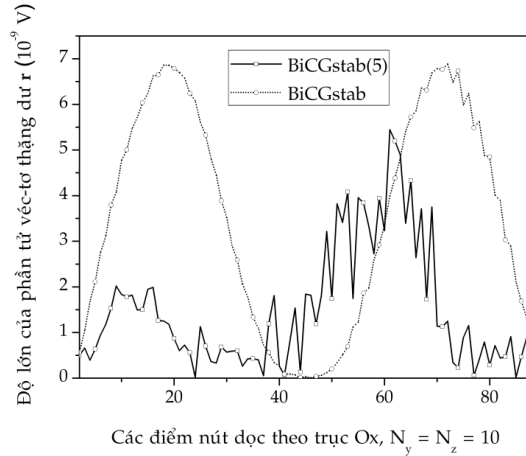
Một trong những điều cần quan tâm khi sử dụng các giải thuật này đó là hiệu năng và thời gian tính toán, thể hiện qua số phép tính toán nhân ma trận (MVS), số lần tính tích nội của hai véc-tơ (DOTS) và số lần cập nhật véc-tơ (AXPYS). Chúng tôi đã thực hiện mô phỏng trên máy tính cá nhân Lenovo Y430 có cấu hình Intel(R) Core(TM)2 Duo CPU T5800 @ 2.00GHz 2.00GHz, DDR II 1.96 GB. Từ Bảng 2 ta thấy rằng khi tăng giá trị của l từ 1 đến 10 trong thuật toán BiCGstab(l) thì MVS, DOTS và AXPYS đều tăng, trong khi số vòng lặp chính cần thực hiện

lại giảm nhanh chóng. Nguyên nhân là ứng với giá trị của l càng lớn thì chi phí tính toán cho một vòng lặp chính càng nhiều. Phép tính MVS tăng theo quy luật ứng với các số tự nhiên chẵn liên tiếp. Mỗi lần l tăng một đơn vị thì số phép tính MVS tăng thêm hai đơn vị. Số phép tính DOTS tăng theo quy luật $0.5(2+7l+l^2)$. Riêng số các phép tính AXPYS tăng nhanh hơn hẳn khi tăng giá trị của l . Trường hợp chương trình giải Poisson có sử dụng thuật toán BiCGstab số phép tính MVS, DOTS và AXPYS ít hơn hẳn khi so với chương trình sử dụng thuật toán BiCGstab(l) với $l = \overline{1,10}$. Trong khi số vòng lặp chính cần thực hiện để tìm nghiệm chính xác cho phương trình Poisson của thuật toán BiCGstab(l) giảm rất nhanh khi tăng dần giá trị của biến l , đồng thời thời gian tính toán cho một lần giải phương trình Poisson lại giảm đáng kể từ 3.6 s ($l = 1$) đến còn 2.6 s ($l = 5$), sau đó thời gian sẽ tăng lên trở lại khi tiếp tục tăng giá trị của l . Điều này cho thấy nếu sử dụng thuật toán BiCGstab(l) thì nên chọn $l = 5$. Tuy nhiên, thuật toán BiCGstab(l) giải phương trình Poisson sẽ chậm hơn 0.43 s khi so sánh với thời gian trung bình giải phương trình Poisson của thuật toán BiCGstab như được chỉ ra trên Hình 6. Để phát huy ưu điểm của cả hai thuật toán này cách tốt nhất là tích hợp cả hai phương pháp trong cùng một lời giải của phương trình Poisson, đó là một hướng mở cho bài toán trong tương lai.

Bảng 2. Số phép tính toán nhân ma trận (MVS), tích nội (DOTS) và cập nhật véc-tơ (AXPYS) của các lời giải phương trình Poisson theo thuật toán BiCGstab, BiCGstab(l) với $l = \overline{1,10}$

	Số vòng lặp chính	MVS	DOTS	AXPYS
BiCGstab	111	3	5	8
BiCGstab(1)	113	2	5	8
BiCGstab(2)	45	4	10	15
BiCGstab(3)	28	6	16	24
BiCGstab(4)	21	8	23	38
BiCGstab(5)	9	10	31	54
BiCGstab(6)	8	12	40	73
BiCGstab(7)	6	14	50	95
BiCGstab(8)	6	16	61	120
BiCGstab(9)	5	18	73	148
BiCGstab(10)	5	20	86	179

Chúng tôi đã thực hiện chương trình mô phỏng ba chiều đi-ốt p-i-n bán dẫn GaAs với chương trình con Poisson sử dụng thuật toán BiCGstab(l) với các giá trị $l = \overline{1, 10}$. So sánh kết quả độ lớn của tất cả các phần tử của véc-tơ thặng dư trong các trường hợp khác nhau của giá trị l và dựa vào Hình 7, chúng tôi rút ra nhận xét rằng với giá trị của $l = 5$ chương trình sẽ cho ra kết quả điện thế với sai số nhỏ nhất, hội tụ nhanh và ổn định nhất.



Hình 7. Đồ thị biểu diễn độ lớn của các thành phần của véc-tơ thặng dư của chương trình con Poisson trong hai trường hợp dùng thuật toán BiCGstab(l) và thuật toán BiCGstab

4 Kết luận

Chúng tôi đã xây dựng thành công một chương trình con Poisson dựa trên thuật toán BiCGstab(l) và gắn vào chương trình mô phỏng ba chiều đi-ốt p-i-n bán dẫn GaAs. Kết quả mô phỏng đã chỉ ra rằng kết quả thu được là phù hợp với các công trình đã công bố trước đây. Chương trình con Poisson dựa trên thuật toán BiCGstab(l) với $l = 5$ sẽ cho kết quả điện thế chính xác hơn với độ lớn của các phần tử của véc-tơ thặng dư nhỏ một cách đồng đều và ổn định hơn. Tuy nhiên thời gian xử lý các tính toán của máy tính lại lớn hơn các chương trình có sử dụng thuật toán BiCGstab đã từng được sử dụng.

Lời cảm ơn

Các tác giả cảm ơn PGS. TS. Đinh Như Thảo, Trường Đại học Sư phạm, Đại học Huế về những góp ý hữu ích cho bản thảo bài báo. Nghiên cứu này được thực hiện với sự tài trợ kinh phí từ đề tài Đại học Huế mã số DHH2017-01-114.

Tài liệu tham khảo

1. D. N. Thao, S. Katayama, and K. Tomizawa (2004), "Numerical simulation of THz radiation by coherent LO phonons in GaAs p-i-n diodes under high electric fields", Journal of Physical Society of Japan, **73**, 11, pp. 3177-3181.
2. D. N. Thảo, D. T. D. My, N. C. P. Thi và N. T. Thủy (2011), "Mô phỏng ba chiều linh kiện na-nô bán dẫn với lời giải phương trình Poisson dựa trên thuật toán GPBICG", Tạp chí Khoa học, Đại học Huế, **65**, pp. 215-223.
3. L. H. Linh (2009), "Mô phỏng động lực học ba chiều của hạt tải trong đi-ốt p-i-n bán dẫn GaAs bằng phương pháp Monte-Carlo", Luận văn Thạc sĩ, Trường Đại học Sư phạm Huế.
4. D. H. Ha et al (2015), "Several efficient algorithms for finding the solution of the three-dimensional Poisson's equation on personal computers", Hue University Journal of Science, **107**, 08, pp. 61-70.
5. K. Tomizawa (1993), "Numerical simulation of submicron semiconductor devices", Artech House, Boston London.
6. H. A. Vorst (2003), "Iterative Krylov methods for large linear systems", Cambridge University.
7. G. L. G. Sleijpen and D. R. Fokkema (1993), "BiCGstab(l) for linear equations involving unsymmetric matrices with complex spectrum", Kent State University, **1**, pp. 11-32.

INVESTIGATING THE SOLUTION OF THE THREE-DIMENSIONAL POISSON'S EQUATION BY USING BiCGstab(l) ALGORITHM

Lê Thị Ngọc Bảo^{1,2}, Dương Đình Phước¹, Lê Quý Thông², Lê Ngọc Minh², Trần Thiện Lân^{1*}

¹HU – University of Education, 34 Le Loi, Hue, Vietnam

²HU – University of Sciences, 77 Nguyen Hue, Hue, Vietnam

Abstract. This paper presents the use of the BiCGstab(l) algorithm to find the solutions for the three-dimensional Poisson's equation. This subroutine was incorporated into the self-consistent ensemble Monte-Carlo device simulation program to test its efficiency. Moreover, in order to compare the convergent rate and stability of the BiCGstab(l) algorithm to the corresponding characteristics of the BiCGstab algorithm, we investigated the dependence of the logarithm of the Euclidean norm of the residual vector on the number of the main loops of Poisson subroutine. The results showed that the program for solving Poisson's equation based on the BiCGstab(l) algorithm gives more accurate simulation results as using the BiCGstab algorithm.

Keywords: Simulation of semiconductor devices, 3D Poisson's equation, BiCGstab(l) algorithm, Monte-Carlo method