Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 754–773
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Numerical schemes for three-dimensional irregular shape quantum dots over curvilinear coordinate systems

Tsung-Min Hwang [a], Wei-Cheng Wang [b], Weichung Wang [c,*]

[a] Department of Mathematics, National Taiwan Normal University, Taipei 116, Taiwan
[b] Department of Mathematics, National Tsing Hua University, Hsinchu 300, Taiwan
[c] Department of Mathematics, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan

Received 24 July 2006; received in revised form 25 April 2007; accepted 28 April 2007
Available online 22 May 2007

Abstract

In this article, we present efficient and stable numerical schemes to simulate three-dimensional quantum dot with irregular shape, so that we can compute all the bound state energies and associated wave functions. A curvilinear coordinate system that fits the target quantum dot shape is first determined. Three finite difference discretizations of the Schrödinger equation are then developed on the original and the skewed curvilinear coordinate system. The resulting large-scale generalized eigenvalue systems are solved by a modified Jacobi–Davidson method. Intensive numerical experiments show that the scheme using both grid points on the original and skewed curvilinear coordinate system can converge to the eigenpairs quickly and stably with second-order accuracy.

Keywords: Three-dimensional irregular shape quantum dot; The Schrödinger equation; Bound state energies and wave functions; Curvilinear coordinate system; Finite difference; Large-scale generalized eigenvalue problem

1. Introduction

Aiming at three-dimensional (3D) semiconductor quantum dots (QD) with irregular shape [3], we intend to develop simple yet efficient and accurate numerical schemes to compute all the bound state energies and the corresponding wave functions of the heterostructure. Numerical simulations of QDs have played an important role for investigating QDs' electronic and optical properties (e.g. [13,16,21]). Among existed methods, finite difference based methods are popular due to their simplicity and possibility to achieve high order convergence rate. However, relatively few numerical schemes focus on 3D settings because of the complicated nature of the QD models. Furthermore, these schemes are usually limited to specific QD geometries like cylinder [10,19],

* Corresponding author. Tel.: +886 2 3366 2810.
E-mail addresses: min@math.ntnu.edu.tw (T.-M. Hwang), wangwc@math.nthu.edu.tw (W.-C. Wang), wwang@math.ntu.edu.tw (W. Wang).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.04.022

cone [11], and pyramid [5,9,14]. To better simulate QDs fabricated in laboratories, simple and efficient numerical schemes that are able to handle 3D irregular QD models would be useful and essential.

We consider a single particle conduction band model that a radial symmetric irregular shape QD is embedded in the center of a cylindrical matrix. Fig. 1 shows a structure scheme of the model. The model is described by the Schrödinger equation:

$$-\nabla\cdot\left(\frac{\hbar^2}{2m(\mathbf{x})}\nabla F\right)+c(\mathbf{x})F=\lambda F, \tag{1}$$

or equivalently, in the cylindrical coordinates,

$$-\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\hbar^2}{2m(\mathbf{x})}\partial_r F\right)-\frac{1}{r}\partial_\theta\left(\frac{1}{r}\frac{\hbar^2}{2m(\mathbf{x})}\partial_\theta F\right)-\partial_z\left(\frac{\hbar^2}{2m(\mathbf{x})}\partial_z F\right)+c(\mathbf{x})F=\lambda F. \tag{2}$$

In Eqs. (1) and (2), $\hbar$ is the reduced Plank constant and the eigenpair (λ, F) stands for the total electron energy and the corresponding wave function. The electron effective mass $m(\mathbf{x})$ and the confinement potential $c(\mathbf{x})$ are material constants and they are piecewise constant functions of the space variable $\mathbf{x}$:

$$m(\mathbf{x})=\begin{cases} m_1 & \text{in the dot,}\\ m_2 & \text{in the matrix,}\end{cases}\qquad c(\mathbf{x})=\begin{cases} c_1 & \text{in the dot,}\\ c_2 & \text{in the matrix.}\end{cases}$$

Associated with the discontinuity in m, we have the following Ben Daniel–Duke interface conditions

$$\begin{aligned} &F|_{D_+}=F|_{D_-},\\ &\frac{\hbar^2}{2m_2}\frac{\partial F}{\partial n}\bigg|_{\partial D_+}=\frac{\hbar^2}{2m_1}\frac{\partial F}{\partial n}\bigg|_{\partial D_-}, \end{aligned} \tag{3}$$

where D is the domain of the QD, n is the normal direction, and the subscripts $+$ and $-$ denote the corresponding outward normal derivatives of the interface that are defined for the matrix and dot regions, respectively. Finally, the Dirichlet condition is applied on the boundary of the cylindrical matrix. For the convenience of further discussions, we define

$$\kappa(\mathbf{x})=\frac{\hbar^2}{2m(\mathbf{x})},$$

and thus

$$\kappa_\ell=\frac{\hbar^2}{2m_\ell},$$

for $\ell=1$ and 2 in the dot and matrix, respectively.

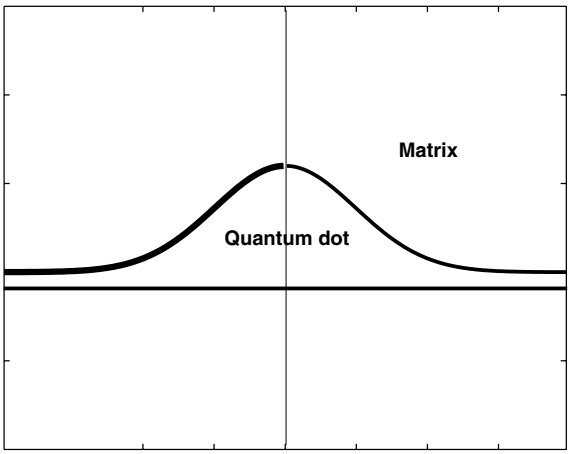

Fig. 1. Structure schema of the quantum dot model.

Several challenges exist for solving the target problem. First, naive discretizations usually result in low accuracy approximation in a irregular shape QD. This is because the interfaces are most likely not aligned with the grid but rather crosses between the grid points, if the Cartesian or cylindrical coordinate is used. In other words, for the grid points around the interfaces, the stencil of a regular finite difference will contain grid points from both sides of the interfaces. Consequently, the standard difference quotients across the interface will not produce accurate approximations to the derivatives, due to the non-smoothness of material parameters. The second difficulty is to solve the large-scale generalized eigenvalue systems arising in the discretizations of the Schrödinger equation in an efficient manner. Since the target eigenvalues are located in the interior of the eigenvalue spectrum, some eigenvalue solvers solve for all the eigenvalues and then identify the target eigenvalues. However, such manners can be extreme time consuming. On the other hand, some dedicated algorithms may be used to find only the target eigenvalues. But such methods may suffer from inefficient preconditioners or numerous non-zero fill-in entries. In short, standard methods like [1,2,6] cannot be recommended due to the inefficiency and instability. See [8] for detail.

To overcome the obstacles, we have accomplished the following results:

- Proposing three discretization schemes based on: (i) the curvilinear coordinate system fitting the irregular interface, (ii) the skewed curvilinear coordinate system induced from the first curvilinear coordinate system, and (iii) the mixed coordinate system integrating the two coordinate systems. The finite difference formulas and related approximation schemes are discussed in detail.
- Modifying the Jacobi–Davidson large-scale eigenvalue solver [8] by suggesting an adaptive scheme for approximating the span vectors. The adaptive scheme carefully incorporates several linear system solvers, preconditioning strategies, and dynamic stopping criteria. The new eigenvalue solver successfully achieves better performances.
- Conducting numerical experiments to justify the schemes' feasibility, efficiency, robustness, and accuracy. The numerical results show that the third scheme developed over both of the original and skewed curvilinear coordinate systems is the quickest and the most stable scheme with second-order convergence rate.

It is worth noting that the main idea of our discretization schemes is motivated from the Jump Condition Capturing Schemes (JCCS) described in [7,20]. The JCCS solves the elliptic interface problem:

$$-\nabla\cdot(\kappa(\mathbf{x})\nabla u(\mathbf{x}))=f(\mathbf{x}),\quad \mathbf{x}\in\Omega\setminus\Gamma, \tag{4}$$

where $\kappa(\mathbf{x})$ is piecewise smooth but discontinuous across a smooth interface $\Gamma\subset\Omega$. Interface conditions $[u]_\Gamma$ and $[\kappa(\mathbf{x})\frac{\partial u}{\partial n}]_\Gamma$ are prescribed in the problem. The notation $[\cdot]_\Gamma$ denotes the difference between the limits from the interior and exterior of the interface Γ. Even though Eqs. (1) and (4) are similar in the sense that they share the same Laplace operator and interface conditions, the discretization of (1) results in eigenvalue systems, while the discretization of (4) leads to linear systems. The difference thus makes the analysis of the eigenvalue systems non-trivial and leads to other computational challenges.

The paper is organized as follows. Section 2 illustrates the curvilinear coordinate systems and three finite difference schemes based on the coordinates. Section 3 presents a modified Jacobi–Davidson method for solving the large-scale generalized eigenvalue systems arising in the discretization schemes. Section 4 presents implementation details and numerical experiments results to justify the performance of the proposed schemes. Section 5 finally concludes the paper.

2. Discretization and finite difference schemes

We develop three finite difference schemes to solve the irregular interface eigenvalue problem (1) in this section. As shown in Fig. 1, the target QD has curve heterojunction structure. It is thus nature to define grid points on a curvilinear coordinate fitting the irregular interface, so that we can avoid using finite difference points that are located on both sides of the heterostructure. To be precise, our first scheme is designed on a curvilinear coordinate system such that some specific constant coordinate curves coincide with the heterostructure interfaces and the matrix boundaries of the QD model. Based on the curvilinear coordinates, our second scheme uses the skewed directions of the previous curvilinear coordinate system which results in sim-

pler formulas. Finally, our third scheme combines the first two schemes by a symmetric averaging. Numerical experiments in Section 4 further show that this method is the quickest and the most stable scheme among the three ones, though this scheme also leads to the most complicated formulas.

We define a curvilinear coordinate system in $\mathbb{R}^n$ as a continuous function $\mathscr{G} : \mathbb{R}^n \to \mathbb{R}^n$. The computation domain U is contained in the domain of $\mathscr{G}$ and the physical domain Ω is covered by the image of $\mathscr{G}$. We assume that the restriction of $\mathscr{G}$ to U is one-to-one, and therefore the inverse $\mathscr{G}^{-1}$ within Ω exists. By such assumption, we can identify points between the physical domain Ω and computation domain U. For example, in $\mathbb{R}^2$, we let ξ^1 and ξ^2 be the variables in the computation domain and $\boldsymbol{x} = (x^1, x^2)$ be the position vector in a 2D physical domain. We then identify $\boldsymbol{x}$ to (ξ^1, ξ^2), if $\boldsymbol{x} = \mathscr{G}(\xi^1, \xi^2)$. The curvilinear coordinates of $\boldsymbol{x}$ determined by $\mathscr{G}$ is $(\xi^1, \xi^2) = \mathscr{G}^{-1}(\boldsymbol{x})$.

We now illustrate how the curvilinear coordinate systems over the physical and computational domains are chosen as shown in Fig. 2. For the QD model specifically, we choose an appropriate curvilinear coordinate system in the following way. Two of physical domain coordinate curves, which are obtained by setting two particular coordinates equal to the constants, coincide with the top interface Γ_T and bottom interface Γ_B of the QD. The exterior boundary of the matrix should also be aligned by certain coordinate curves. A logical mesh is then formed by the intersections of the coordinate lines in a certain curvilinear coordinate system. For simplicity, we assume the computational grid points are distributed uniformly along each of the coordinates and have the step sizes $\Delta\xi^1$ and $\Delta\xi^2$ in the ξ^1 and ξ^2 directions, respectively. The set of two number $(i\Delta\xi^1, k\Delta\xi^2)$, for $i = 1, \ldots, M$ and $k = 0, \ldots, N$, is a typical grid point in the computation domain. The curvilinear coordinate system should also fit the QD heterojunction in top and bottom by, for example, $\xi^2 = N_2\Delta\xi^2$ and $\xi^2 = N_1\Delta\xi^2$, for a certain N_2 and N_1 between 0 and N.

Note that we do not require the coordinate system to be orthogonal in the physical domain. For the convenience, we use the pairing (i, k) to indicate the point which has the discrete curvilinear coordinates $(i\Delta\xi^1, k\Delta\xi^2)$. We also use $f_{i,k}$ to indicate the value of the function f at the point $\boldsymbol{x}(i\Delta\xi^1, k\Delta\xi^2)$.

Based on the chosen curvilinear coordinate system, Eq. (2) can be re-written as

$$\frac{-1}{r}\frac{1}{\sqrt{\mathbf{g}}}\sum_{\alpha,\beta=1}^{2}\partial_{\xi^\alpha}\left(r\kappa\sqrt{\mathbf{g}}\mathbf{g}^{\alpha\beta}\partial_{\xi^\beta}F\right) - \frac{\kappa}{r^2}\partial_\theta^2 F + cF = \lambda F, \tag{5}$$

where the metric tensors are defined by

$$\mathbf{g}^{\alpha\beta} := \left\langle \nabla\xi^\alpha, \nabla\xi^\beta \right\rangle, \tag{6}$$

$$\mathbf{g}_{\alpha\beta} := \left\langle \frac{\partial \boldsymbol{x}}{\partial\xi^\alpha}, \frac{\partial \boldsymbol{x}}{\partial\xi^\beta} \right\rangle, \tag{7}$$

and

$$\mathbf{g} := \det\begin{pmatrix} \mathbf{g}_{11} & \mathbf{g}_{12} \\ \mathbf{g}_{21} & \mathbf{g}_{22} \end{pmatrix}. \tag{8}$$

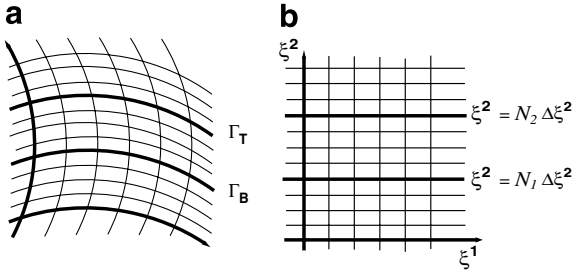

Fig. 2. Curvilinear coordinate systems for the physical and computational domains are shown in Parts (a) and (b), respectively. The irregular heterojunctions are fitted by the curves Γ_T and Γ_B as shown in (a). The curves are mapped to the corresponding lines $\xi^2 = N_2\Delta\xi^2$ and $\xi^2 = N_1\Delta\xi^2$ in (b).

The notation $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product. We note that, by Eqs. (7) and (8), $\sqrt{\mathbf{g}}$ is the Jacobian determinant of $\boldsymbol{x}$ with respect to $\boldsymbol{\xi}$, i.e.,

$$\sqrt{\mathbf{g}} = \det\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right). \tag{9}$$

Furthermore,

$$\sum_{\iota=1}^{2} \mathbf{g}^{\alpha\iota}\mathbf{g}_{\iota\beta} = \delta^{\alpha}_{\beta}, \tag{10}$$

for $\alpha, \beta = 1, 2$. Detail descriptions of vector analysis over curvilinear coordinate systems can be found in, for example [4].

While the QD model can be described by Eq. (5) in three-dimensional on the curvilinear coordinates, we may rewrite the equation to transform the 3D problem to a sequence of 2D problems and gain significant savings in computation. Since the solution F is periodic in θ direction, we can approximate it by the truncated Fourier series as

$$F(\xi^1, \xi^2, \theta) = \sum_{n=-L/2}^{L/2-1} \widehat{F}^n(\xi^1, \xi^2)\mathrm{e}^{\mathrm{i}n\theta},$$

where $\widehat{F}^n(\xi^1, \xi^2)$ is the complex Fourier coefficient given by

$$\widehat{F}^n(\xi^1, \xi^2) = \frac{1}{L}\sum_{j=0}^{L-1} F(\xi^1, \xi^2, \theta_j)\mathrm{e}^{-\mathrm{i}n\theta_j},$$

for $\theta_j = 2j\pi/L$, and L is the number of grid points along the θ-direction. Substituting the above expansions into Eq. (5) and equating the Fourier coefficients, we derive $\widehat{F}^n$ satisfying the 2D Schrödinger equations

$$\frac{-1}{r}\frac{1}{\sqrt{\mathbf{g}}}\sum_{\alpha,\beta=1}^{2} \partial_{\xi^\alpha}(r\kappa\sqrt{\mathbf{g}}\mathbf{g}^{\alpha\beta}\partial_{\xi^\beta}\widehat{F}^n) + \left(c + \frac{n^2}{r^2}\kappa\right)\widehat{F}^n = \lambda\widehat{F}^n,$$

or

$$-\sum_{\alpha,\beta=1}^{2} \partial_{\xi^\alpha}(r\kappa\sqrt{\mathbf{g}}\mathbf{g}^{\alpha\beta}\partial_{\xi^\beta}\widehat{F}^n) + r\sqrt{\mathbf{g}}\left(c + \frac{n^2}{r^2}\kappa\right)\widehat{F}^n = r\sqrt{\mathbf{g}}\lambda\widehat{F}^n, \tag{11}$$

for a certain Fourier mode n. By using the above transformation, we only need to solve several 2D problems for a certain Fourier modes to obtain all the eigenpairs that are of interest. Note that usually the number of 2D problems needed to be solved is much less than N. Similar dimension reduction techniques have been applied to the cylindrical quantum dot models in [19]. In that paper, however, the authors derive the formulas from the viewpoints of matrix analysis.

2.1. The first scheme: curvilinear coordinate system based

Now we derive our first finite difference scheme for solving the 2D problems (11). The finite difference formulas are carefully formulated by using the grid points located on the chosen curvilinear coordinate system, so that the scheme can achieve the second-order convergence rate numerically.

We first define the difference and averaging operators as follows:

$$(\mathscr{D}_{\xi^1}\psi)_{i,k} = \frac{1}{\Delta\xi^1}\left(\psi_{i+\frac{1}{2},k} - \psi_{i-\frac{1}{2},k}\right),$$
$$(\mathscr{D}_{\xi^2}\psi)_{i,k} = \frac{1}{\Delta\xi^2}\left(\psi_{i,k+\frac{1}{2}} - \psi_{i,k-\frac{1}{2}}\right),$$

$$(\mathscr{A}_1\psi)_{i,k} = \frac{1}{2}\left(\psi_{i+\frac{1}{2},k} + \psi_{i-\frac{1}{2},k}\right),$$
$$(\mathscr{A}_2\psi)_{i,k} = \frac{1}{2}\left(\psi_{i,k+\frac{1}{2}} + \psi_{i,k-\frac{1}{2}}\right).$$

Using these operators, the summation

$$\sum_{\alpha,\beta=1}^{2} \partial_{\xi^\alpha}(r\kappa\sqrt{\mathbf{g}}\mathbf{g}^{\alpha\beta}\partial_{\xi^\beta}\widehat{F}^n)$$

in the first term of Eq. (11) at grid point (i, k) is approximated by

$$(L_{11}\widehat{F}^n)_{i,k} + (L_{12}\widehat{F}^n)_{i,k} + (L_{21}\widehat{F}^n)_{i,k} + (L_{22}\widehat{F}^n)_{i,k}, \tag{12}$$

where

$$(L_{11}\widehat{F}^n)_{i,k} = \mathscr{D}_{\xi^1}(\mathscr{A}_2(r\sqrt{g_h}\kappa g_h^{11})\mathscr{D}_{\xi^1}\widehat{F}^n)_{i,k}, \tag{13}$$
$$(L_{22}\widehat{F}^n)_{i,k} = \mathscr{D}_{\xi^2}(\mathscr{A}_1(r\sqrt{g_h}\kappa g_h^{22})\mathscr{D}_{\xi^2}\widehat{F}^n)_{i,k}, \tag{14}$$
$$(L_{12}\widehat{F}^n)_{i,k} = \mathscr{D}_{\xi^1}(\mathscr{A}_2(r\sqrt{g_h}\kappa g_h^{12}\mathscr{A}_1\mathscr{D}_{\xi^2}\widehat{F}^n))_{i,k}, \tag{15}$$
$$(L_{21}\widehat{F}^n)_{i,k} = \mathscr{D}_{\xi^2}(\mathscr{A}_1(r\sqrt{g_h}\kappa g_h^{21}\mathscr{A}_2\mathscr{D}_{\xi^1}\widehat{F}^n))_{i,k}, \tag{16}$$

and g_h and $g_h^{\alpha\beta}$ denote the approximations of the metric tensors $\mathbf{g}$ and $\mathbf{g}^{\alpha\beta}$, respectively. Note that the full expressions for $L_{11}\hat{F}^n$, $L_{22}\widehat{F}^n$, $L_{12}\hat{F}^n$, and $L_{21}\hat{F}^n$ can be found in Appendix.

Based on the formulas discussed above, Eq. (11) is approximated by various c and κ as shown below. The choices of c and κ are depended on the locations of the central grid points, $\boldsymbol{x}(i, k)$ or $\boldsymbol{x}(\xi_i^1, \xi_k^2)$. For a certain index k, $c = c_1$ and $\kappa = \kappa_1$, if the (virtual) grid point corresponding to $k + \frac{1}{2}$ belongs to the dot. Similarly, $c = c_2$ and $\kappa = \kappa_2$, if $k + \frac{1}{2}$ belongs to the matrix.

$$-\sum_{\alpha,\beta=1}^{2}(L_{\alpha\beta}\widehat{F}^n)_{i,k} + \left\{\frac{r_i}{4}(\mathscr{A}_2\mathscr{A}_1\sqrt{g_h}c)_{i,k} + \frac{n^2}{4r_i}(\mathscr{A}_2\mathscr{A}_1\sqrt{g_h}\kappa)_{i,k}\right\}\widehat{F}_{i,k}^n = \frac{r_i}{4}(\mathscr{A}_2\mathscr{A}_1\sqrt{g_h})_{i,k}\lambda\widehat{F}_{i,k}^n. \tag{17}$$

Finally, by suitably combining all the finite difference schemes discussed above, we obtain the resulting generalized eigenvalue system under the (computational) curvilinear coordinate system:

$$A_L\widehat{F} = \lambda B_L\widehat{F}, \tag{18}$$

where $\widehat{F}$ is the unknown vector containing the wave function on the grid points, A_L contains nine non-zero components in every row, and B_L is a diagonal matrix. It is worth mentioning that matrix A_L in (18) is symmetric and positive definite, thanks to the approximation scheme defined in Eqs. (13)–(16) as shown in [20]. Furthermore, eigenvalue solvers can be benefit from this property in the sense of efficiency and stability.

2.2. Skewed coordinate system

Alternatively, we consider the equation on a skewed curvilinear coordinate system as shown in Fig. 3. To be specific, variables of the skewed coordinate system (η^1, η^2) are defined by

$$\eta^1 := \frac{\xi^1\Delta\xi^2 + \xi^2\Delta\xi^1}{\sqrt{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}}, \tag{19}$$
$$\eta^2 := \frac{\xi^2\Delta\xi^1 - \xi^1\Delta\xi^2}{\sqrt{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}}. \tag{20}$$

This definition also leads to the fact that the step sizes on the η^1 and η^2 directions are identical; that is,

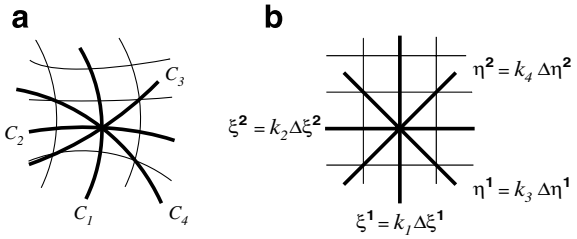

Fig. 3. Schema of the skewed curvilinear coordinate system is shown in part (b), while the physical domain coordinate system is shown in (a). In (a), the curves C_1 and C_2 are mapped to the lines $\xi^1 = k_1\Delta\xi^1$ and $\xi^2 = k_2\Delta\xi^2$ in the curvilinear coordinate system, respectively. Similarly, the curves C_3 and C_4 are mapped to the lines $\eta^1 = k_3\Delta\eta^1$ and $\eta^2 = k_4\Delta\eta^2$ in the skewed curvilinear coordinate system, respectively.

$$\Delta\eta^1 = \Delta\eta^2 = \frac{2\Delta\xi^1\Delta\xi^2}{\sqrt{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}} \equiv \Delta\eta.$$

The idea of using the skewed curvilinear coordinate system is motivated by [20]. In that paper, the author develops finite difference formulas on the skewed curvilinear coordinates to solve the elliptic interface problem (4). The scheme results in the linear systems that are consistent, symmetric, and positive definite. Such matrix structures obviously favor certain linear system solvers. It is still open whether similar conclusions, regarding the matrix structures and the favor in eigenvalue solvers, can be drawn while the resulting scheme (28) is applied. However, the finite difference scheme based on the skewed curvilinear coordinate does lead to simpler formulas and a more sophisticated hybrid scheme in Section 2.3.

The equivalent formulation of the 2D Schrödinger equation (11) on the skewed curvilinear coordinate (η^1, η^2) for a certain mode n can be represented as

$$-\sum_{\mu,\nu=1}^{2} \partial_{\eta^\mu}(r\kappa\sqrt{\hat{\mathbf{g}}}\hat{\mathbf{g}}^{\mu\nu}\partial_{\eta^\nu}\widehat{F}^n) + r\sqrt{\hat{\mathbf{g}}}\left(c + \frac{n^2}{r^2}\kappa\right)\widehat{F}^n = r\sqrt{\hat{\mathbf{g}}}\lambda\widehat{F}^n,$$

or

$$\frac{-1}{r}\frac{1}{\sqrt{\hat{\mathbf{g}}}}\sum_{\mu,\nu=1}^{2} \partial_{\eta^\mu}(r\kappa\sqrt{\hat{\mathbf{g}}}\hat{\mathbf{g}}^{\mu\nu}\partial_{\eta^\nu}\widehat{F}^n) + \left(c + \frac{n^2}{r^2}\kappa\right)\widehat{F}^n = \lambda\widehat{F}^n, \tag{21}$$

where the metric tensor with respect to the skewed variables (η^1, η^2) is given by

$$\hat{\mathbf{g}}^{\mu\nu} := \langle\nabla\eta^\mu, \nabla\eta^\nu\rangle, \tag{22}$$

$$\hat{\mathbf{g}}_{\mu\nu} := \left\langle\frac{\partial \boldsymbol{x}}{\partial\eta^\mu}, \frac{\partial \boldsymbol{x}}{\partial\eta^\nu}\right\rangle, \tag{23}$$

$$\hat{\mathbf{g}} := \det\begin{pmatrix}\hat{\mathbf{g}}_{11} & \hat{\mathbf{g}}_{12}\\ \hat{\mathbf{g}}_{21} & \hat{\mathbf{g}}_{22}\end{pmatrix}. \tag{24}$$

An obvious conclusion is that

$$\sqrt{\hat{\mathbf{g}}} = \det\left(\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\eta}}\right) = \varphi\sqrt{\mathbf{g}}, \quad \text{where } \varphi = \frac{1}{2}\left(\frac{\Delta\xi^1}{\Delta\xi^2} + \frac{\Delta\xi^2}{\Delta\xi^1}\right) \tag{25}$$

and

$$\sum_{\gamma=1}^{2} \hat{\mathbf{g}}^{\mu\gamma}\hat{\mathbf{g}}_{\gamma\nu} = \delta_\nu^\mu. \tag{26}$$

Now we approximate Eq. (21) by second-order centered differences. By defining the skewed difference operations

$$(\mathscr{D}_{\eta^1}\psi)_{i,k} = \frac{1}{\Delta\eta^1}\left(\psi_{i+\frac{1}{2},k+\frac{1}{2}} - \psi_{i-\frac{1}{2},k-\frac{1}{2}}\right)$$

and

$$(\mathscr{D}_{\eta^2}\psi)_{i,k} = \frac{1}{\Delta\eta^2}\left(\psi_{i-\frac{1}{2},k+\frac{1}{2}} - \psi_{i+\frac{1}{2},k-\frac{1}{2}}\right),$$

the summation

$$\sum_{\mu,\nu=1}^{2} \partial_{\eta^\mu}(r\kappa\sqrt{\hat{\mathbf{g}}}\hat{\mathbf{g}}^{\mu\nu}\partial_{\eta^\nu}\widehat{F}^n)$$

in the first term of Eq. (21) at grid point (i,k) is approximated by

$$\sum_{\mu,\nu=1}^{2} \mathscr{D}_{\eta^\mu}(r\sqrt{\hat{g}_h}\kappa\hat{g}_h^{\mu\nu}\mathscr{D}_{\eta^\nu}\widehat{F}^n)_{i,k} \equiv S^+(\kappa,i,k) - S^-(\kappa,i,k),$$

where

$$\begin{aligned}
S^+(i,k,\kappa) = \frac{1}{(\Delta\eta)^2}\Big(& (r\sqrt{\hat{g}_h}\kappa\hat{g}_h^{11})_{i+\frac{1}{2},k+\frac{1}{2}}(\widehat{F}^n_{i+1,k+1} - \widehat{F}^n_{i,k}) + (r\sqrt{\hat{g}_h}\kappa\hat{g}_h^{12})_{i+\frac{1}{2},k+\frac{1}{2}}(\widehat{F}^n_{i,k+1} - \widehat{F}^n_{i+1,k}) \\
& + (r\sqrt{\hat{g}_h}\kappa\hat{g}_h^{21})_{i-\frac{1}{2},k+\frac{1}{2}}(\widehat{F}^n_{i,k+1} - \widehat{F}^n_{i-1,k}) + (r\sqrt{\hat{g}_h}\kappa\hat{g}_h^{22})_{i-\frac{1}{2},k+\frac{1}{2}}(\widehat{F}^n_{i-1,k+1} - \widehat{F}^n_{i,k})\Big), \\
S^-(i,k,\kappa) = \frac{1}{(\Delta\eta)^2}\Big(& (r\sqrt{\hat{g}_h}\kappa\hat{g}_h^{11})_{i-\frac{1}{2},k-\frac{1}{2}}(\widehat{F}^n_{i,k} - \widehat{F}^n_{i-1,k-1}) + (r\sqrt{\hat{g}_h}\kappa\hat{g}_h^{12})_{i-\frac{1}{2},k-\frac{1}{2}}(\widehat{F}^n_{i-1,k} - \widehat{F}^n_{i,k-1}) \\
& + (r\sqrt{\hat{g}_h}\kappa\hat{g}_h^{21})_{i+\frac{1}{2},k-\frac{1}{2}}(\widehat{F}^n_{i+1,k} - \widehat{F}^n_{i,k-1}) + (r\sqrt{\hat{g}_h}\kappa\hat{g}_h^{22})_{i+\frac{1}{2},k-\frac{1}{2}}(\widehat{F}^n_{i,k} - \widehat{F}^n_{i+1,k-1})\Big),
\end{aligned}$$

and $\hat{\mathrm{g}}_h$ and $\hat{g}_h^{\mu\nu}$ denote the approximations of the metric tensors $\hat{\mathbf{g}}$ and $\hat{\mathbf{g}}^{\mu\nu}$, respectively.

Consequently, Eq. (21) can be approximated by Eq. (27) in the following manners. For a certain index k of the central grid points (i,k), $c = c_1$ and $\kappa = \kappa_1$, if the (virtual) grid point corresponding to $k+\frac{1}{2}$ belongs to the dot. Similarly, $c = c_2$ and $\kappa = \kappa_2$, if $k+\frac{1}{2}$ belongs to the matrix:

$$\begin{aligned}
& -S^+(i,k,\kappa) + S^-(i,k,\kappa) + \left\{\frac{r_i}{4}(\mathscr{A}_2\mathscr{A}_1\sqrt{\hat{g}_h}c)_{i,k} + \frac{n^2}{4r_i}(\mathscr{A}_2\mathscr{A}_1\sqrt{\hat{g}_h}\kappa)_{i,k}\right\}\widehat{F}^n_{i,k} \\
& \quad = \frac{r_i}{4}\left(\mathscr{A}_2\mathscr{A}_1\sqrt{\hat{g}_h}\right)_{i,k}\lambda\widehat{F}^n_{i,k}.
\end{aligned} \tag{27}$$

Combining above finite difference scheme under the skewed curvilinear coordinate system, the resulting generalized eigenvalue system to be solved becomes

$$A_S\widehat{F} = \lambda B_S\widehat{F}, \tag{28}$$

where A_S is a symmetric positive definite matrix with nine non-zero components in every row and B_S is a diagonal matrix.

2.3. Mixed scheme

Now we consider a mixed scheme combining the finite difference schemes developed in Sections 2.1 and 2.2. The idea of the mixed scheme is inspired by the discretization scheme of the elliptic operator $\nabla\cdot(\kappa(\mathbf{x})\nabla)$ proposed by Huang and Wang in [7], in which the scheme is developed to solve the elliptic interface problems with jump conditions in the form of (4). Further, it has been shown that under certain conditions: (i) the resulting discretization matrix is symmetric and positive definite, (ii) the method is monotone preserving (i.e. the coefficient matrix in the resulting linear system is a M-matrix), and (iii) the scheme has the second-order accuracy. As the elliptic interface problems (4) and the Schrödinger equation (1) have similar elliptic operator, we thus suggest applying the idea of mixed discretization scheme to the problem considered here with following

remark. It is not clear whether the mixed method has the second-order accuracy analytically while applying in the eigenvalue problem (1). In our numerical experiments, however, the mixed method does achieve the second-order convergence numerically and performs stably in the sense that all target eigenpairs have been obtained.

We combine the two schemes defined in (11) and (21) to form a new mixed scheme by using the symmetry preserving average. First, we show that Eq. (29) is actually equivalent to the original target problem (1). Consider

$$\varphi(-\sqrt{\mathbf{g}}\nabla\cdot(\omega\kappa\nabla\mathbf{F})+\omega\sqrt{\mathbf{g}}\mathbf{c}\mathbf{F})+\left(-\sqrt{\hat{\mathbf{g}}}\nabla\cdot((\mathbf{1}-\omega)\kappa\nabla\mathbf{F})+(\mathbf{1}-\omega)\sqrt{\hat{\mathbf{g}}}\mathbf{c}\mathbf{F}\right)=\lambda\sqrt{\hat{\mathbf{g}}}\mathbf{F}, \tag{29}$$

where ω is a constant that $0\leqslant\omega\leqslant 1$. Since $\sqrt{\hat{\mathbf{g}}}=\varphi\sqrt{\mathbf{g}}$ (see Eq. (25)), we can divide both sides of Eq. (29) by $\sqrt{\hat{\mathbf{g}}}$ to obtain

$$\omega(-\nabla\cdot(\kappa\nabla F)+cF)+(1-\omega)(-\nabla\cdot(\kappa\nabla F)+cF)=\lambda F, \tag{30}$$

or

$$-\nabla\cdot(\kappa\nabla F)+cF=\lambda F,$$

which is exactly the same equation as the target problem (1). Note that the derivation above holds similarly provided ω is a location depended weight function.

Without loss of generality, we take $\Delta\xi^1=\Delta\xi^2$ (and therefore $\varphi=1$) to obtain the following equation:

$$\omega(-\sqrt{\hat{\mathbf{g}}}\nabla\cdot(\kappa\nabla F)+\sqrt{\hat{\mathbf{g}}}cF)+(1-\omega)\left(-\sqrt{\hat{\mathbf{g}}}\nabla\cdot(\kappa\nabla F)+\sqrt{\hat{\mathbf{g}}}cF\right)=\lambda\sqrt{\hat{\mathbf{g}}}F. \tag{31}$$

By further applying the Fourier transformation and other techniques discussed in the previous sub-sections, Eq. (31) can be approximated by the corresponding discretization schemes discussed in previous sections. To be specific, the resulting eigenvalue system becomes

$$A_{\mathrm{M}}\widehat{F}^n=\lambda B_{\mathrm{M}}\widehat{F}^n, \tag{32}$$

where $A_{\mathrm{M}}=\omega A_L+(1-\omega)A_{\mathrm{S}}$, $B_{\mathrm{M}}=B_L=B_{\mathrm{S}}$, and ω is a constant function.

Remark. Though it has been shown in [7] that there exist a weight function ω such that the discretization of the mixed scheme is monotone for the problem described in (4), how ω affect the eigenvalue problem (1) is again unclear analytically. However, we have investigated numerically the effect of the weight function ω and then make a suggestion of choosing the weight function to increase the efficiency of the mixed scheme in Section 4.2.

2.4. Approximations of matrix tensors

In this section, we present the approximations of the matrix tensors used in the discretization schemes. Note that similar matrix tensor approximations are considered in [7].

Let $\hat{g}^h_{\mu,\nu}$ denotes the second-order approximation of the metric tensor $\hat{\mathbf{g}}_{\mu\nu}$ in the skewed curvilinear coordinate (η^1,η^2). By the definition of $\hat{\mathbf{g}}_{\mu\nu}$ in Eq. (23), $\hat{g}^h_{\mu,\nu}$ can be easily computed by the difference quotients of the position vectors $\boldsymbol{x}_{ij}$'s as follows:

$$(\hat{g}^h_{11})_{i+\frac{1}{2},k+\frac{1}{2}}:=\frac{\mathbf{x}_{i+1,k+1}-\mathbf{x}_{i,k}}{\Delta\eta^1}\cdot\frac{\mathbf{x}_{i+1,k+1}-\mathbf{x}_{i,k}}{\Delta\eta^1},$$

$$(\hat{g}^h_{22})_{i+\frac{1}{2},k+\frac{1}{2}}:=\frac{\mathbf{x}_{i,k+1}-\mathbf{x}_{i+1,k}}{\Delta\eta^2}\cdot\frac{\mathbf{x}_{i,k+1}-\mathbf{x}_{i+1,k}}{\Delta\eta^2},$$

$$(\hat{g}^h_{12})_{i+\frac{1}{2},k+\frac{1}{2}}=(\hat{g}^h_{21})_{i+\frac{1}{2},k+\frac{1}{2}}:=\frac{\mathbf{x}_{i+1,k+1}-\mathbf{x}_{i,k}}{\Delta\eta^1}\cdot\frac{\mathbf{x}_{i,k+1}-\mathbf{x}_{i+1,k}}{\Delta\eta^2}.$$

It should be noted that these quantities are all evaluated at cell centers. By Eq. (24), the numerical analogous formula is

$$\hat{g}_h = \det\begin{pmatrix} \hat{g}^h_{11} & \hat{g}^h_{12} \\ \hat{g}^h_{21} & \hat{g}^h_{22} \end{pmatrix}.$$

According to Eq. (26), the numerical quantities $\hat{g}^{\mu\nu}_h$'s can be obtained by solving the linear system

$$\sum_{\iota=1}^{2} \hat{g}^{\mu\iota}_h \hat{g}^h_{\iota\nu} = \delta^\mu_\nu,$$

that is, for $\mu,\nu = 1,2$,

$$\begin{pmatrix} \hat{g}^h_{11} & \hat{g}^h_{12} \\ \hat{g}^h_{21} & \hat{g}^h_{22} \end{pmatrix}\begin{pmatrix} \hat{g}^{11}_h & \hat{g}^{12}_h \\ \hat{g}^{21}_h & \hat{g}^{22}_h \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Let $g^{\alpha\beta}_h$ be the second-order approximation of the metric tensor $\mathbf{g}^{\alpha\beta}$ in the ordinary curvilinear coordinate (ξ^1, ξ^2). Then, inspired by (25), we define

$$\sqrt{g_h} := \frac{2\Delta\xi^1\Delta\xi^2}{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}\sqrt{\hat{g}_h}.$$

From the definitions of $\mathbf{g}^{\alpha\beta}$, $\hat{\mathbf{g}}^{\mu\nu}$ and (η^1, η^2) in Eqs. (6), (22) and (19), (20), respectively, we have

$$\mathbf{g}^{11} = \frac{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}{4(\Delta\xi^2)^2}(\hat{\mathbf{g}}^{11} + \hat{\mathbf{g}}^{22} - 2\hat{\mathbf{g}}^{12}),$$

$$\mathbf{g}^{22} = \frac{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}{4(\Delta\xi^1)^2}(\hat{\mathbf{g}}^{11} + \hat{\mathbf{g}}^{22} + 2\hat{\mathbf{g}}^{12}),$$

$$\mathbf{g}^{12} = \mathbf{g}^{21} = \frac{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}{4\Delta\xi^1\Delta\xi^2}(\hat{\mathbf{g}}^{11} - \hat{\mathbf{g}}^{22}).$$

Therefore, we define

$$(g^{11}_h)_{i+\frac{1}{2},k+\frac{1}{2}} := \frac{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}{4(\Delta\xi^2)^2}(\hat{g}^{11}_h + \hat{g}^{22}_h - 2\hat{g}^{12}_h)_{i+\frac{1}{2},k+\frac{1}{2}},$$

$$(g^{22}_h)_{i+\frac{1}{2},k+\frac{1}{2}} := \frac{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}{4(\Delta\xi^1)^2}(\hat{g}^{11}_h + \hat{g}^{22}_h + 2\hat{g}^{12}_h)_{i+\frac{1}{2},k+\frac{1}{2}},$$

$$(g^{12}_h)_{i+\frac{1}{2},k+\frac{1}{2}} = (g^{21}_h)_{i+\frac{1}{2},k+\frac{1}{2}} := \frac{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}{4\Delta\xi^1\Delta\xi^2}(\hat{g}^{11}_h - \hat{g}^{22}_h)_{i+\frac{1}{2},k+\frac{1}{2}}.$$

3. The eigenvalue problem solver

In Section 2, we have discussed three finite difference schemes aiming at solving the Schrödinger equation. These schemes lead to the large-scale generalized eigenvalue systems in the form of (18), (28) and (32). To solve these eigenvalue problems, we use the Jacobi–Davidson method (JDM) sketched in Fig. 4. The preference of the JDM is based on the successful experience on solving the eigenvalue systems arising in various QD models detailed in [8,9,19].

While the framework of the JDM used in this paper is similar to the ones in [8,9,19], we can further improve the algorithm by solving the correct equation

$$\left(I - \frac{p_k u_k^*}{u_k^* p_k}\right)(A - \theta_k B)\left(I - \frac{u_k p_k^*}{u_k^* p_k}\right)t = -r_k \tag{33}$$

in Step (v) of Fig. 4 by using preconditioning iterative approximations as suggested in [17]. The method uses a preconditioner

Given a vector v_1 with $v_1^* B v_1 = 1$; set $V_1 = [v_1]$.
Compute $W_1 = AV_1$ and $M_1 = V_1^* W_1$.
Do $k = 1, 2, 3, \ldots$
(i) Compute all the eigenpairs of the small size problem $(M_k - \theta I_k)s = 0$.
(ii) Select the desired (target) eigenvalue θ_k and let s_k be the associated normalized eigenvector.
(iii) Compute $u_k = V_k s_k$, $p_k = Bu_k$, and $r_k = (A - \theta_k B)u_k$.
(iv) If $(||r_k||_2 < \varepsilon)$, Set $\lambda = \theta_k$, $x = u_k$, Stop.
(v) Solve (approximately) a $t \perp_B u_k$ from $(I - \frac{p_k u_k^*}{u_k^* p_k})(A - \theta_k B)(I - \frac{u_k p_k^*}{u_k^* p_k})t = -r_k$.
(vi) B-Orthogonalize t against V_k; Set $v_{k+1} = t / \| t \|_B$.
(vii) Compute $w_{k+1} = Av_{k+1}$, $M_{k+1} = \begin{pmatrix} M_k & V_k^* w_{k+1} \\ v_{k+1}^* W_k & v_{k+1}^* w_{k+1} \end{pmatrix}$,
(viii) Expand $W_{k+1} = [W_k, w_{k+1}]$ and $V_{k+1} = [V_k, v_{k+1}]$

Fig. 4. The Jacobi–Davidson method for computing the target eigenvalues and the associated eigenvectors of generalized eigenvalue problems $Ax - \lambda Bx = 0$.

$$\mathscr{M}_p \equiv \left(I - \frac{p_k u_k^*}{u_k^* p_k}\right)\mathscr{M}\left(I - \frac{u_k p_k^*}{u_k^* p_k}\right) \approx \left(I - \frac{p_k u_k^*}{u_k^* p_k}\right)(A - \theta_k B)\left(I - \frac{u_k p_k^*}{u_k^* p_k}\right),$$

where $\mathscr{M}$ is an approximation of $(A - \theta_k B)$, and an iterative method to solve Eq. (33). In each of the iterative steps, the solution of the linear system

$$\mathscr{M}_p t = y$$

is computed as follows for a certain given vector y:

$$t = \mathscr{M}^{-1} y - \zeta \mathscr{M}^{-1} B u_k, \tag{34}$$

where

$$\zeta = \frac{u_k^{\mathrm{T}} B \mathscr{M}^{-1} y}{u_k^{\mathrm{T}} B \mathscr{M}^{-1} B u_k}.$$

While there are various iterative methods available and many parameters allowing tuning up, we adopt the following heuristic strategies to gain better timing efficiency while solving Eq. (33).

These heuristic strategies are determined by the numerical experience and adaptive residual stopping criteria. The iterative methods are stopped if the residual of the linear systems are less than the accuracy requirement that are set to be gradually higher and higher as the iterations go on. The heuristics are specified by the linear system solver GMRES [15] and Bi-CGSTAB [18], preconditioning strategy SSOR [6], maximum iteration number, and residual stopping criterion. For example, {`GMRES`, `SSOR`, `30`, $\mathtt{10}^{-3}$} indicates that GMRES with SSOR preconditioner is used to solve a linear system, the maximum iteration number is 30, and the residual stopping criterion is 10^{-3}. The specific heuristic strategies used to solve the linear system (33) are described as follows:

- To compute the first smallest positive eigenvalue (λ_1): in such cases, Step (v) of the algorithm in Fig. 4 is changed to the pseudo-code illustrated in Fig. 5.
- To compute the other positive eigenvalues ($\lambda_2, \lambda_3, \ldots$): in such cases, Step (v) of the algorithm in Fig. 4 is changed to the pseudo-code illustrated in Fig. 6. Note that the variable j is set to be equal to 15 initially.

(v) Solve Eq. (33) (approximately) to obtain a $t \perp_B u_k$ by the method determined below.
If ($k \leq 9$) then
Use {BiCGSTAB, No precond., 7, 10^{-3}}
else
Use {GMRES, SSOR, 30, 10^{-3}}
End if

Fig. 5. The heuristic strategy for computing the smallest positive eigenvalue.

(v) Solve Eq. (33) (approximately) to obtain a $t \perp_B u_k$ by the method determined below.
If ($\| r_k \|_2 > 0.1$ and $k < 10$) then
Use {BiCGSTAB, No precond., 7, 10^{-3}}
else if ($\| r_k \|_2 \geq 0.1$ and $k > 14$) then
Use {GMRES, SSOR, 30, 10^{-3}}
else if ($\| r_k \|_2 < 0.1$ and $\| r_{k-1} \|_2 / \| r_k \|_2 < 4$) then
Set $j = \min(30, j + 2)$ and use {GMRES, SSOR, j, 10^{-3}}
else
Use {GMRES, SSOR, j, 10^{-3}}
End if

Fig. 6. The heuristic strategy for computing eigenvalues other than the smallest positive one.

4. Computational results

We have implemented the proposed schemes and conducted numerical experiments to evaluate the efficiency, accuracy, and convergence behavior of the schemes. We first describe the implementation details in Section 4.1 and then demonstrate the performance results in Section 4.2.

The numerical experiments are conducted on a workstation running HP Unix. The workstation is equipped with an Intel 1.3 GHz Itanium II CPU and 24-gigabyte main memory. The simulation (811×2390 grid points) takes up about 1.2-gigabyte main memory.

The proposed schemes are implemented by Fortran 90 and compiled by the HP Fortran Compiler with options `+DSitanium2 +O2 +U77 +DD64 +Ofast +Onolimit`.

4.1. Implementation details

First of all, we describe how we choose the grid points in the numerical experiments. The grid points in the 3D physical domain are denoted as $(r_{i,j,k}, \theta_{i,j,k}, z_{i,j,k})$, for $i = 1, \ldots, M$, $j = 1, \ldots, L$, and $k = 0, \ldots, N$. For computational purposes, however, we only need to consider the discretization scheme on the 2D hyperplanes spanned by the radial and axial directions. This is because the target 3D problem can be decoupled into a sequence of independent 2D problems as shown in Section 2.1. Therefore, we use the abbreviations $(r_{i,k}, z_{i,k})$ to denote the grid points for a certain hyperplane with a fixed θ.

The grid points are chosen uniformly along the radial direction, with a modification that the grid points are shifted with a half mesh width [12]. The grid points located on the top and bottom of the QD are determined by user defined smooth shape functions in the physical domain that fit the QD shape to be simulated. Grid points in zone $\mathscr{Z}_A$, $\mathscr{Z}_B$, and $\mathscr{Z}_C$ (as shown in Fig. 7) are chosen by the following means. The grid points in zone $\mathscr{Z}_B$ are chosen uniformly along the axial axis. In zone $\mathscr{Z}_A$ and $\mathscr{Z}_C$, however, the grid points are chosen

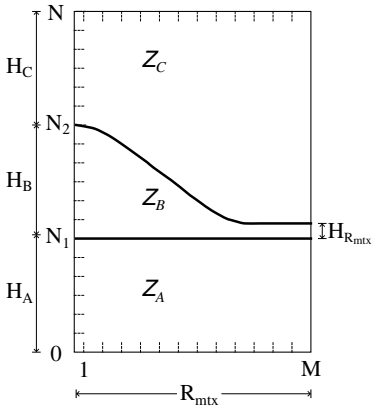

Fig. 7. Discretization schema of a 2D half plane. Grid points are indexed from 1 to M in the radial coordinate and 0 to N in the axial coordinate. The grid points located on the two curves that fit the bottom and the top of the QD are indexed as N_1 and N_2, respectively. The r–z hyperplane is divided into three zones: $\mathscr{Z}_A$ (below the QD), $\mathscr{Z}_B$ (interior of the QD), and $\mathscr{Z}_C$ (above the QD). The heights of $\mathscr{Z}_A$, $\mathscr{Z}_B$, and $\mathscr{Z}_C$ in the center are denoted as H_A, H_B and H_C, respectively. The height of the wetting layer in the matrix boundary is denoted by $H_{R_{mtx}}$.

non-uniformly in a way that fine meshes are created around the heterojunction. Such non-uniform mesh scheme allows us to capture rapid changes of wave functions around the heterojunction. Fig. 8 illustrates a schema of the grid points on the r–z hyperplane. As shown in Fig. 8, the chosen grid points are usually condensed more than necessary in the thin wetting layer of the QD. We adopt this particular grid distribution on the physical domain simply for programming convenience, rather than for accuracy nor efficiency considerations. It can be seen that the resulting computational domain is a logical rectangle with uniform mesh and two coordinate lines (in physical space) aligned with the interfaces. Consequently, the uniform lattice in the computational domain can significantly ease the computer programming efforts. However, it is worth noting that the computational cost and convergence rate for both uniform and distorted grids are actually similar based on our numerical experiments.

Taking our implementation as an example, the grid points along the radial direction are

$$r_{i,k} = r(\xi_i^1, \xi_k^2) = \left(i - \frac{1}{2}\right)\Delta r,$$

where $\Delta r = \frac{R_{mtx}}{M-\frac{1}{2}}$ for all i and k. By using the notations introduced in Fig. 7, we define $z_{i,k}$'s as follows. The grid points located on the bottom of the QD are specified by the (linear) shape function:

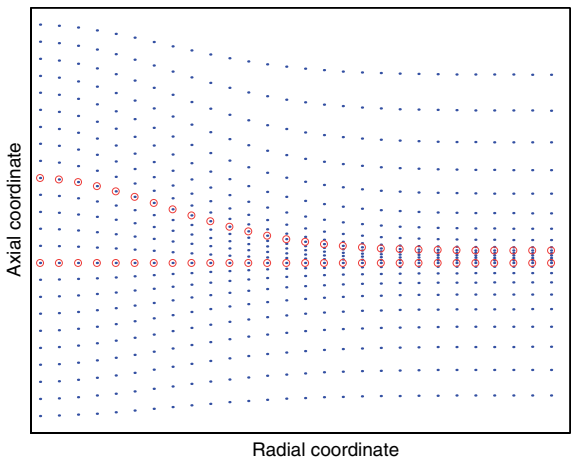

Fig. 8. A schema of the grid points on the r–z hyperplane. The circled grid points indicate the interfaces of the quantum dot.

$$z_{i,N_1} = z(\xi_i^1, \xi_{N_1}^2) = N_1 h_z,$$

where h_z is a length constant defined as

$$h_z = \frac{H_B}{N_2 - N_1} = \frac{H_A}{N_1} = \frac{H_C}{N - N_2}$$

for $i = 1, \ldots, M$. The grid points located on the top of the QD are determined by the (nonlinear) shape function in the physical domain:

$$z_{i,N_2} = z(\xi_i^1, \xi_{N_2}^2) = \tau_{top}(r_i),$$

where

$$\tau_{top}(r) = a + (H_A + H_B - a)e^{-(r/R_{mtx})^2 \times (4096/500)} \tag{35}$$

and

$$a = H_A + H_{R_\infty} \tag{36}$$

for $i = 1, \ldots, M$. Here, H_{R_∞} is a reference height that will be specified below.

The grid points in other parts are defined as follows:

$$z_{i,k} = z(\xi_i^1, \xi_k^2) = \begin{cases} \ell(i,k) - q_1(i,k), & \text{for } 1 \leqslant k \leqslant N_1, \quad (\text{in } \mathscr{Z}_A) \\ \ell(i,k), & \text{for } N_1 \leqslant k \leqslant N_2, \quad (\text{in } \mathscr{Z}_B) \\ \ell(i,k) + q_2(i,k), & \text{for } N_2 \leqslant k \leqslant N, \quad (\text{in } \mathscr{Z}_C) \end{cases} \tag{37}$$

where

$$\ell(i,k) = N_1 h_z + \frac{\tau_{top}(r_i) - N_1 h_z}{(N_2 - N_1)h_z}(k - N_1)h_z,$$

$$q_1(i,k) = \ell(i,0)\left(\frac{N_1 - k}{N_1}\right)^2,$$

$$q_2(i,k) = (z_N - \ell(i,N))\left(\frac{k - N_2}{N - N_2}\right)^2.$$

Note that Eq. (37) can be realized from another viewpoint as follows. For a fixed i, $z_{i,k}$ is constructed as a C^1 function of k which is linear in Z_B and quadratic in Z_A and Z_C.

Finally, we specify some details of our implementation. The structure parameters illustrated in Fig. 7 are: $H_A = 16.4$ nm, $H_B = 8.2$ nm, $H_C = 16.4$ nm, $H_{R_\infty} = 8.2/7$ nm, and $R_{mtx} = 14$ nm.

The effective electron mass for the QD and the matrix are $0.024m_e$ and $0.067m_e$, respectively. The confinement potential $c_1 = 0.000$ eV and $c_2 = 0.770$ eV. The iteration process within the Jacobi–Davidson subroutine terminates when the absolute error of the residual in the eigenvalue problems less than 1.0×10^{-10} for the computed eigenpairs.

4.2. Performance of the schemes

Denoting the three schemes discussed in Sections 2.1, 2.2 and 2.3 as S_c (on curvilinear coordinates), S_s (on skewed curvilinear coordinates), and S_m (mixed scheme), respectively, we now present the performances of the schemes in terms of the computation of eigenpairs, timing results, and convergence rates. Note that schemes S_s and S_c are the special cases of S_m that $\omega = 0$ and 1, respectively.

4.2.1. Computed eigenpairs

We first focus on the results of computed eigenpairs. While the three schemes are able to compute all the physical meaningful bound state energies, scheme S_s does produce spurious eigenpairs. Furthermore, the eigenvectors corresponding to these redundant eigenvalues are oscillatory. Fig. 9 shows the spectrums of the computed eigenvalues. It is clear that all of the three schemes result in similar target eigenvalues, while

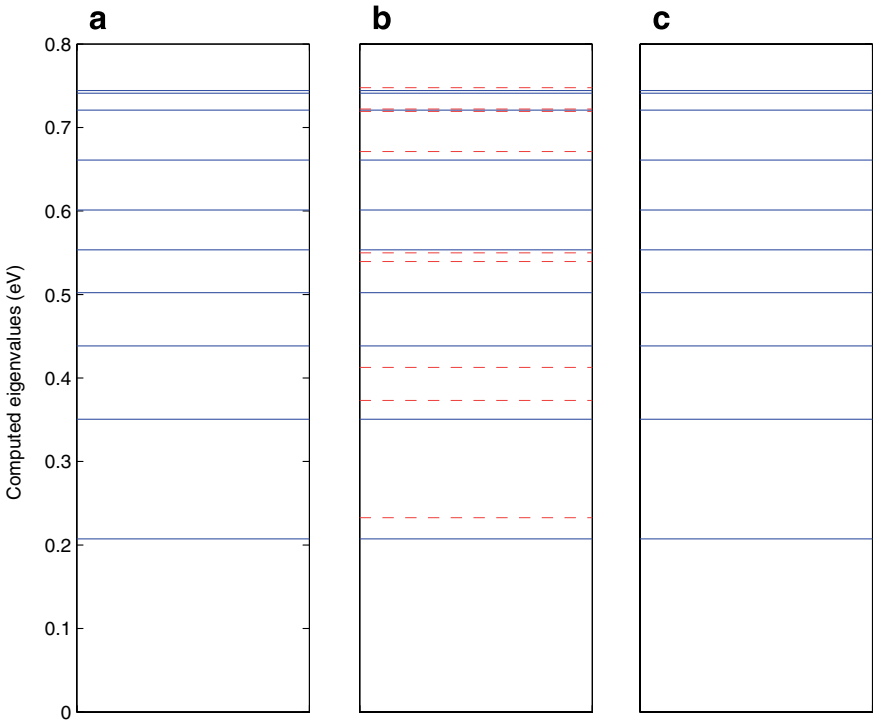

Fig. 9. The spectrums of energy levels (eigenvalues) computed by schemes S_c, S_s, and S_m ($\omega = 0.1$) are shown in part (a), (b), and (c), respectively. Scheme S_s has produced several extra eigenvalues due to the discretization scheme. These spurious eigenvalues are plotted by dashed lines in part (b).

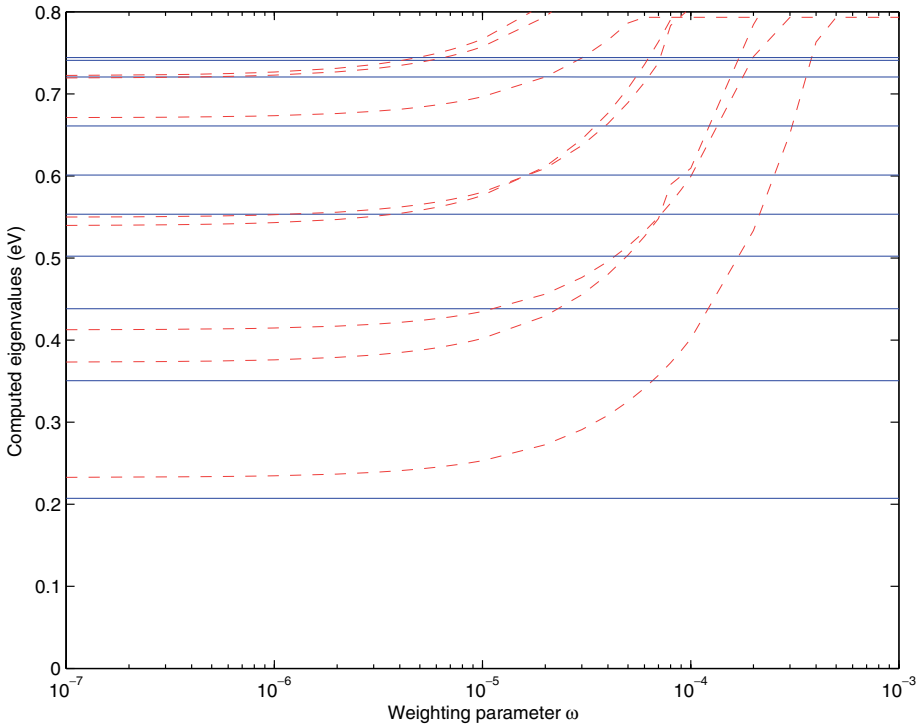

Fig. 10. The values of spurious eigenvalues corresponding to the weighting parameters $10^{-7} < \omega < 0.001$ are plotted by dashed lines. These spurious eigenvalues increase to the values that are larger than the energy gap (0.770 eV) rapidly.

scheme S_s leads to several extra computed eigenvalues. It is worth noting that these spurious eigenvalues do not simply vanish whenever scheme S_m is used for $\omega > 0$. Fig. 10 shows how these spurious eigenvalues increase to the values that are larger than the energy gap 0.770 eV (i.e. beyond the target region) for tiny ω's. The figure also suggests that even a relatively small weighting parameter (e.g. $\omega = 0.001$) enables scheme S_m to avoid obtaining spurious eigenvalues. Fig. 11 further shows the eigenvector corresponding to the spurious eigenvalue (0.2327 eV) produced by Scheme S_s. It is clear that this wave function (eigenvector), and other eigenvectors corresponding to the spurious eigenvalues, are oscillatory. Such spurious eigenvalue and oscillatory eigenvector behavior of S_s, while solving Eq. (21), is due to the fact that no Dirichlet boundary condition is imposed on $r = 0$ and thus the Black–Red decoupling occurs.

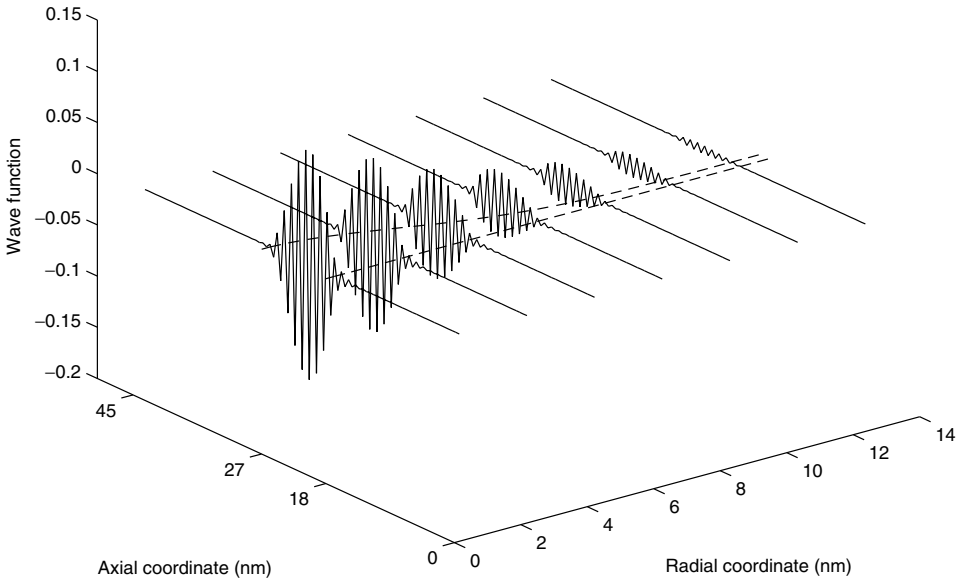

Fig. 11. The eigenvector corresponding to the spurious eigenvalue (0.2327 eV) produced by Scheme S_s is oscillatory. The eigenvector is plotted over the r–z hyperplane. For clearness, only the eigenvector values on the a certain axial lines are presented here. The quantum dot is demarcated by the dashed line.

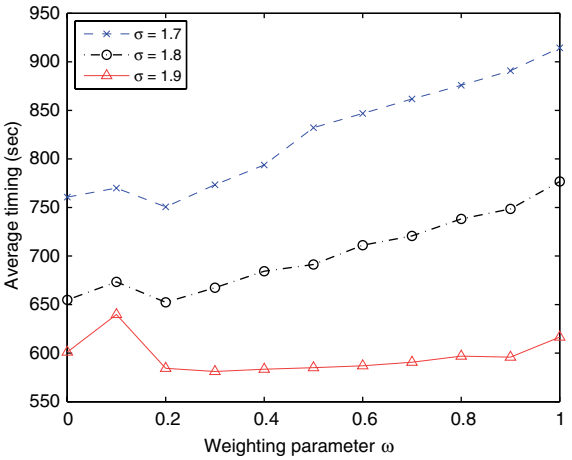

Fig. 12. Average timing results for computing all the target eigenvalues. Scheme S_s ($\omega = 0$), S_c ($\omega = 1$), and S_m ($\omega = 0.1$–0.9) are compared by using three different SSOR preconditioner parameter σ that is equal to 1.7, 1.8, and 1.9. The results suggest that the mixed scheme S_m that $\omega = 0.2$ is the best choice generally.

4.2.2. Timing results

Fig. 12 shows average timing results of S_m for computing all the target eigenpairs by letting SSOR preconditioner parameter $\sigma = 1{,}7$, 1.8, and 1.9. In the figure, the x-axis represents the weight coefficient ω defined in Eq. (31). The matrix dimensions of the eigensystem is $811 \times 2390 = 1{,}935{,}090$. It is obvious that scheme S_m for $\omega = 0.2$ generally outperforms other cases in timing, no matter what value of σ is. Furthermore, the figure suggests that the optimal choice of σ is 1.9 and the choice leads to efficient timing results for a wide range of ω's.

4.2.3. Convergence rates

We demonstrate convergence rates of the three schemes while computing all the target eigenvalues. To measure the convergence rate, we run the tests over the meshes described in Table 1 and then calculate the rates by

$$\text{rate}_{[i]} = \log_3\left(\frac{\lambda_{[i]} - \lambda_{[i+1]}}{\lambda_{[i+1]} - \lambda_{[i+2]}}\right), \quad \text{for } i = 1, \ldots, 3,$$

where $\lambda_{[i]}$ for $i = 1, \ldots, 3$ denote the approximate eigenvalues obtained from the meshes described in the table. Fig. 13 illustrates the convergence rates, $\text{rate}_{[1]}$, $\text{rate}_{[2]}$, and $\text{rate}_{[3]}$, for all target eigenvalues computed by the

Table 1
Dimension information regarding the computation of convergence rates

(M, N)	Matrix dimension $(M-1)\cdot(N-1)$	Eigenvalue	Rate of convergence
(21, 60)	1180	$\lambda_{[1]}$	–
(62, 180)	10,919	$\lambda_{[2]}$	–
(185, 540)	99,176	$\lambda_{[3]}$	$\text{rate}_{[1]}$
(554, 1620)	895,307	$\lambda_{[4]}$	$\text{rate}_{[2]}$
(1661, 4860)	8,065,940	$\lambda_{[5]}$	$\text{rate}_{[3]}$

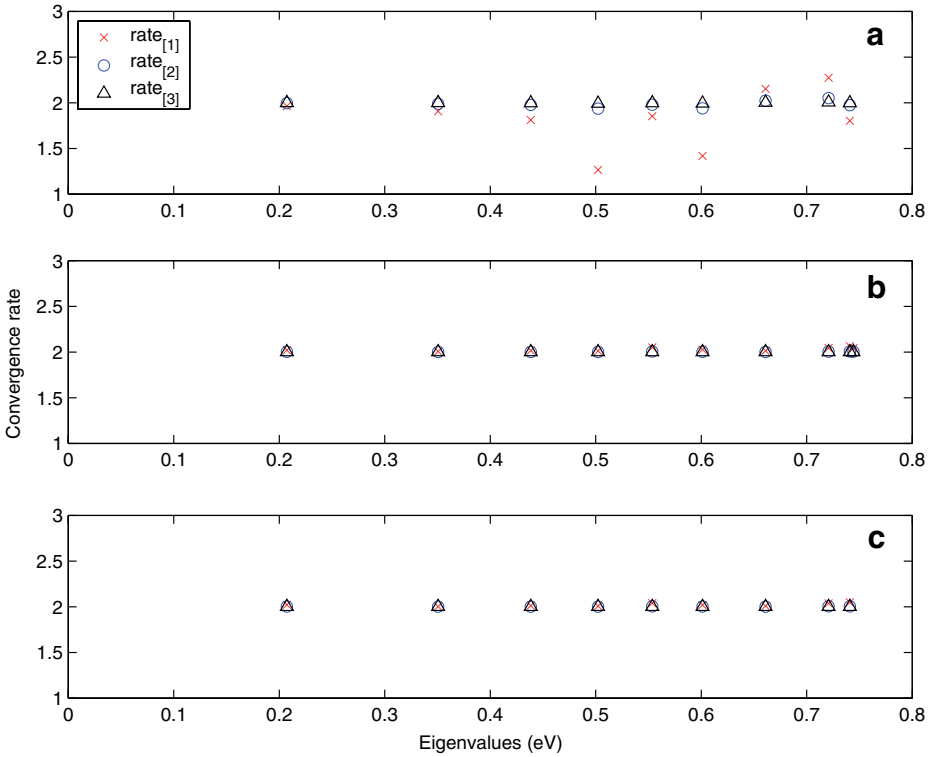

Fig. 13. The eigenvalue convergence rates for schemes S_c, S_s, and S_m are shown in part (a), (b) and (c), respectively.

Table 2
Summary of the proposed numerical schemes

Scheme	Formula	Spurious eigenpair	Timing	Conv. rate
S_c	In between	None	Slow	≈2 (with variation)
S_s	Simple	Several	Quick	≈2
S_m	Complicated	None	Quickest (for $\omega = 0.1$)	≈2

schemes S_c, S_s, and S_m. From the figure, we see that all the convergence rates of S_s and S_m are close to 2. For S_c, the rate$_{[2]}$'s and rate$_{[3]}$'s are close to 2; however, several rate$_{[1]}$'s are less than 2. In short, all three schemes achieve second-order convergence rates for fine grids. Furthermore, schemes S_s and S_m have also achieved second-order convergence rate steadily, even for loose grids.

The characteristics of the three numerical schemes are summarized in Table 2. Observing the table, we conclude that, with the price of more complicated formulas, Scheme S_m converges to the eigenpairs quickest, achieves second-order convergence rate consistently, and does not result in any spurious eigenpairs.

5. Conclusion

This article aims at developing efficient, robust, and accurate numerical schemes to simulate quantum dots with irregular dot shape. We have proposed three second-order finite difference schemes to discretize the Schrödinger equation. The first scheme is developed naturally on a curvilinear coordinate system fitting the shape of the quantum dot. The second scheme uses the skewed curvilinear coordinate system and leads to simple formulas. The third scheme combines the two previous schemes by a symmetric averaging. The resulting large-scale generalized linear eigenvalue problems are solved efficiently by a modified Jacobi–Davidson method. Intensive numerical experiments have shown that the mixed scheme converges to the eigenpairs quickest, achieves second-order convergence rate consistently, and does not result in any spurious eigenpairs. While the numerical experiments have shown very promising results, theoretical analysis of the discretizations and the resulting eigenvalue problems are still open.

Acknowledgements

We are grateful to Wen-Wei Lin and Yin-Liang Huang for many helpful discussions with them and to Yi-Hsien Liu for assistance in some programming efforts and computational results. We are also grateful to the referees for their valuable suggestions and comments. This work is partially supported by the National Science Council and the National Center for Theoretical Sciences in Taiwan.

Appendix

By the definitions of the operators $\mathscr{D}_{\xi^1}$, $\mathscr{D}_{\xi^2}$, $\mathscr{A}_1$, and $\mathscr{A}_2$, Eqs. (13)–(16) can be represented as

$$(L_{11}\widehat{F}^n)_{i,k} = \frac{1}{\Delta\xi^1}\left\{\frac{1}{2}\left((r\sqrt{g_h}\kappa g_h^{11})_{i+\frac{1}{2},k+\frac{1}{2}} + (r\sqrt{g_h}\kappa g_h^{11})_{i+\frac{1}{2},k-\frac{1}{2}}\right)\cdot\left(\frac{\widehat{F}^n_{i+1,k}-\widehat{F}^n_{i,k}}{\Delta\xi^1}\right)\right\}$$
$$-\frac{1}{\Delta\xi^1}\left\{\frac{1}{2}\left((r\sqrt{g_h}\kappa g_h^{11})_{i-\frac{1}{2},k+\frac{1}{2}} + (r\sqrt{g_h}\kappa g_h^{11})_{i-\frac{1}{2},k-\frac{1}{2}}\right)\cdot\left(\frac{\widehat{F}^n_{i,k}-\widehat{F}^n_{i-1,k}}{\Delta\xi^1}\right)\right\},$$

$$(L_{22}\widehat{F}^n)_{i,k} = \frac{1}{\Delta\xi^2}\left\{\frac{1}{2}\left((r\sqrt{g_h}\kappa g_h^{22})_{i+\frac{1}{2},k+\frac{1}{2}} + (r\sqrt{g_h}\kappa g_h^{22})_{i-\frac{1}{2},k+\frac{1}{2}}\right)\cdot\left(\frac{\widehat{F}^n_{i,k+1}-\widehat{F}^n_{i,k}}{\Delta\xi^2}\right)\right\}$$
$$-\frac{1}{\Delta\xi^2}\left\{\frac{1}{2}\left((r\sqrt{g_h}\kappa g_h^{22})_{i+\frac{1}{2},k-\frac{1}{2}} + (r\sqrt{g_h}\kappa g_h^{22})_{i-\frac{1}{2},k-\frac{1}{2}}\right)\cdot\left(\frac{\widehat{F}^n_{i,k}-\widehat{F}^n_{i,k-1}}{\Delta\xi^2}\right)\right\},$$

$$
\begin{aligned}
(L_{12}\widehat{F}^n)_{i,k} = & \frac{1}{\Delta\xi^1}\frac{1}{2}\left\{(r\sqrt{g_h}\kappa g_h^{12})_{i+\frac{1}{2},k+\frac{1}{2}}\left(\frac{1}{2}\left(\frac{\widehat{F}^n_{i,k+1}-\widehat{F}^n_{i,k}}{\Delta\xi^2}+\frac{\widehat{F}^n_{i+1,k+1}-\widehat{F}^n_{i+1,k}}{\Delta\xi^2}\right)\right)\right.\\
& \left.+(r\sqrt{g_h}\kappa g_h^{12})_{i+\frac{1}{2},k-\frac{1}{2}}\left(\frac{1}{2}\left(\frac{\widehat{F}^n_{i,k}-\widehat{F}^n_{i,k-1}}{\Delta\xi^2}+\frac{\widehat{F}^n_{i+1,k}-\widehat{F}^n_{i+1,k-1}}{\Delta\xi^2}\right)\right)\right\}\\
& -\frac{1}{\Delta\xi^1}\frac{1}{2}\left\{(r\sqrt{g_h}\kappa g_h^{12})_{i-\frac{1}{2},k+\frac{1}{2}}\left(\frac{1}{2}\left(\frac{\widehat{F}^n_{i-1,k+1}-\widehat{F}^n_{i-1,k}}{\Delta\xi^2}+\frac{\widehat{F}^n_{i,k+1}-\widehat{F}^n_{i,k}}{\Delta\xi^2}\right)\right)\right.\\
& \left.+(r\sqrt{g_h}\kappa g_h^{12})_{i-\frac{1}{2},k-\frac{1}{2}}\left(\frac{1}{2}\left(\frac{\widehat{F}^n_{i-1,k}-\widehat{F}^n_{i-1,k-1}}{\Delta\xi^2}+\frac{\widehat{F}^n_{i,k}-\widehat{F}^n_{i,k-1}}{\Delta\xi^2}\right)\right)\right\},\\
(L_{21}\widehat{F}^n)_{i,k} = & \frac{1}{\Delta\xi^2}\frac{1}{2}\left\{(r\sqrt{g_h}\kappa g_h^{21})_{i+\frac{1}{2},k+\frac{1}{2}}\left(\frac{1}{2}\left(\frac{\widehat{F}^n_{i+1,k+1}-\widehat{F}^n_{i,k+1}}{\Delta\xi^1}+\frac{\widehat{F}^n_{i+1,k}-\widehat{F}^n_{i,k}}{\Delta\xi^1}\right)\right)\right.\\
& \left.+(r\sqrt{g_h}\kappa g_h^{21})_{i-\frac{1}{2},k+\frac{1}{2}}\left(\frac{1}{2}\left(\frac{\widehat{F}^n_{i,k+1}-\widehat{F}^n_{i-1,k+1}}{\Delta\xi^1}+\frac{\widehat{F}^n_{i,k}-\widehat{F}^n_{i-1,k}}{\Delta\xi^1}\right)\right)\right\}\\
& -\frac{1}{\Delta\xi^2}\frac{1}{2}\left\{(r\sqrt{g_h}\kappa g_h^{21})_{i+\frac{1}{2},k-\frac{1}{2}}\left(\frac{1}{2}\left(\frac{\widehat{F}^n_{i+1,k}-\widehat{F}^n_{i,k}}{\Delta\xi^1}+\frac{\widehat{F}^n_{i+1,k-1}-\widehat{F}^n_{i,k-1}}{\Delta\xi^1}\right)\right)\right.\\
& \left.+(r\sqrt{g_h}\kappa g_h^{21})_{i-\frac{1}{2},k-\frac{1}{2}}\left(\frac{1}{2}\left(\frac{\widehat{F}^n_{i,k}-\widehat{F}^n_{i-1,k}}{\Delta\xi^1}+\frac{\widehat{F}^n_{i,k-1}-\widehat{F}^n_{i-1,k-1}}{\Delta\xi^1}\right)\right)\right\}.
\end{aligned}
$$

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, third ed., SIAM, Philadelphia, 1999.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, 2000.

[3] D.M. Bruls, P.M. Koenraad, H.W.M. Salemink, J.H. Wolter, M. Hopkinson, M.S. Skolnick, Stacked low-growth-rate inas quantum dots studied at the atomic level by cross-sectional scanning tunneling microscopy, Applied Physics Letters 82 (21) (2003) 3758–3760.

[4] H. Davis, Introduction to Vector Analysis, second ed., Allyn and Bacon, 1967.

[5] D. El-Moghraby, R.G. Johnson, P. Harrison, Calculating modes of quantum wire and dot systems using a finite differencing technique, Computer Physics Communications 150 (2003) 235–246.

[6] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, 1996.

[7] Y.-L. Huang, W.-C. Wang, A monotone jump condition capturing scheme for elliptic interface problems on irregular domains. Preprint.

[8] T.-M. Hwang, W.-W. Lin, J.-L. Liu, W. Wang, Jacobi–Davidson methods for cubic eigenvalue problems, Numerical Linear Algebra with Applications 12 (2005) 605–624.

[9] T.-M. Hwang, W.-W. Lin, W.-C. Wang, W. Wang, Numerical simulation of three dimensional pyramid quantum dot, Journal of Computational Physics 196 (2004) 208–232.

[10] T.-M. Hwang, W. Wang, Numerical studies of energy states in vertically aligned quantum dot array, Computers and Mathematics with Applications 49 (2005) 39–51.

[11] Y. Li, O. Voskoboynikov, C.P. Lee, S.M. Sze, Computer simulation of electron energy levels for different shape InAs/GaAs semiconductor quantum dots, Computer Physics Communications (141) (2001) 66–72.

[12] P.E. Merilees, The pseudo-spectral approximation applied to the shallow water equations on a sphere, Atmosphere 11 (1973) 13–20.

[13] F.M. Peeters, V.A. Schweigert, Two-electron quantum disks, Physics Review B 53 (1996) 1468–1474.

[14] C. Pryor, Eight-band calculations of strained InAs/GaAs quantum dots compared with one-, four-, and six-band approximations, Physics Review B 57 (1998) 7190–7195.

[15] Y. Saad, M. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (1986) 856–869.

[16] J. Shumway, L.R.C. Fonseca, J.P. Leburton, R.M. Martin, D.M. Ceperley, Electronic structure of self-assembled quantum dots: comparison between density functional theory and diffusion quantum Monte Carlo, Physica E 8 (2000) 260–268.

[17] G.L.G. Sleijpen, H.A. van der Vorst, A Jacobi–Davidson iteration method for linear eigenvalue problems, SIAM Journal on Matrix Analysis and Applications 17 (2) (1996) 401–425.

[18] H.A. Van Der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 13 (1992) 631–644.
[19] W. Wang, T.-M. Hwang, W.-W. Lin, J.-L. Liu, Numerical methods for semiconductor heterostructures with band nonparabolicity, Journal of Computational Physics 190 (1) (2003) 141–158.
[20] W.-C. Wang, A jump condition capturing finite difference scheme for elliptic interface problems, SIAM Journal of Scientific Computing 25 (5) (2004) 1479–1496.
[21] A.J. Williamson, A. Zunger, InAs quantum dots: predicted electronic structure of free-standing versus GaAs-embedded structures, Physics Review B 59 (1999) 15819–15824.