Kinetic and Related Models
©American Institute of Mathematical Sciences
Volume **4**, Number **4**, December **2011**
doi:10.3934/krm.2011.4.1097
pp. **1097–1120**

A HYBRID SCHRÖDINGER/GAUSSIAN BEAM SOLVER FOR QUANTUM BARRIERS AND SURFACE HOPPING

Shi Jin and Peng Qi

Department of Mathematics, University of Wisconsin
480 Lincoln Drive, Madison, WI 53706, USA

Abstract. In this paper, we propose a hybrid method coupling a Schrödinger solver and the Gaussian beam method for the numerical simulation of quantum tunneling through potential barriers or surface hopping across electronic potential energy surfaces. The idea is to use a Schrödinger solver near potential barriers or zones where potential energy surfaces cross, and the Gaussian beam method–which is much more efficient than a direct Schrödinger solver–elsewhere. Buffer zones are used to convert data between the Schrödinger solver and the Gaussian beam solver. Numerical examples show that this method indeed captures quantum tunneling and surface hopping accurately, with a computational cost much lower than a direct quantum solver in the entire domain.

1. **Introduction.** The fundamental equation in quantum mechanics is the Schrödinger equation,

$$\begin{aligned} i\varepsilon\psi\left(\mathbf{x},\,t\right)_t &= -\frac{\varepsilon^2}{2}\triangle\psi\left(\mathbf{x},\,t\right)+V\left(\mathbf{x}\right)\psi\left(\mathbf{x},\,t\right), \\ \psi\left(\mathbf{x},\,0\right) &= \psi_0\left(\mathbf{x}\right), \end{aligned} \tag{1}$$

where $\psi\left(\mathbf{x},\,t\right)$ is the wave function, $V\left(\mathbf{x}\right)$ the potential, and ε a small parameter, which is typically the reduced Planck constant.

It is well known that computation at the quantum level is prohibitively expensive, even for the one-body Schrödinger equation, due to the existence of the small parameter ε which characterizes the quantum scale. The solution to the Schrödinger equation is oscillatory with wave lengths of order $O\left(\varepsilon\right)$. One needs $O(\varepsilon)$ mesh sizes to represent the wave. The time step should also be $O(\varepsilon)$ if the wave function is to be computed accurately, even with the best spectral method. See [2]. When the potential is discontinuous, one is subject to even more severe mesh size and time constraints. See [13].

The Gaussian beam method (GB) [6, 9, 10, 28, 30] is an asymptotic method to solve the Schrödinger equation, allowing a computational mesh of size $O(\sqrt{\varepsilon})$, which is much less expensive than a direct Schrödinger solver. Although more expensive than the geometric optics approach based on classical mechanics, GB offers accurate solutions of density at caustics, as well as phase information. It has drawn many

2000 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases. Gaussian beam, Time-splitting spectral method, Surface hopping, quantum tunneling.

This work was partially supported by NSF grant DMS-0608720 and NSF FRG grant DMS-0757285. SJ was also supported by a Van Vleck Distinguished Research Prize and a Vilas Associate Award from the University of Wisconsin-Madison.

recent activities, see for examples [17, 23, 24, 25, 34]. GB can also be extended to handle interface/barriers, see [40, 37, 41, 39]. For a recent comprehensive survey of semi-classical methods for the Schrödinger equation, see [12].

In this paper, we are interested in computing two quantum tunneling phenomena: 1) tunneling through a potential barrier; and 2) surface hopping between electronic potential energy surfaces. We are interested in a general method that can be applied to different barriers or potential energy surfaces. For problem 1) there have been semi-classical methods developed using quantum scattering information, see [4, 13, 14, 15]. In the latter case, the surface hopping method, pioneered by Tully, [32], has seen flourishing development [33, 38, 31, 7, 20, 21, 22, 11, 16]. The Landau-Zener formula [19, 43] that provides the transition probability between potential energy surfaces is often used in a surface hopping method. The hybrid method proposed in this paper, that hybridizes a Schrödinger solver near the barriers or hopping zones with GB elsewhere, allows one to treat general barriers or hopping surfaces, and capture the phase information accurately both inside and outside the barriers or surface hopping areas.

In a relevant work, a hybrid method combining the finite difference method and GB for high frequency wave propagations was developed in [36]. In this method, the whole space is divided into areas according to whether the wave speed changes fast or not. The wave equation is first written into a hyperbolic system. In the area where the wave speed varies slowly, the system is diagonalized and GB is used for the decoupled system. In the area where the wave speed changes fast, GB is not proper. The Gaussian beams are then converted into a wave function on a local mesh and evolved using a local finite difference method. When the wave finally leaves the fast speed changing area, the wave function is turned back into Gaussian beams using the method developed in [35]. This hybrid method reduces the computational cost for this multi-scale problem.

We adopt the same framework here, with several distinct features and different applications. First, we use a fixed mesh, rather than a local moving mesh as in [36], to solve the Schrödinger equation (using the time-splitting spectral method [2]). We then use two buffer zones around each of the barriers or surface hopping areas to convert the data from Schrödinger to Gaussian beams, and vice versa. This is important since the conversion goes only one way (namely once, say the Schrödinger data is converted into Gaussian beams, it will not be converted back to Schrödinger in the same buffer zone). It is to the best of our knowledge that this is the first hybrid method for the surface hopping problems.

The paper is organized as follows. In Section 2, we present a system of Schrödinger equations for surface hopping that arises in the Born-Oppenheimer approximation [3]. In Section 3, we propose our hybrid algorithm for the Schrödinger equations with discontinuous potentials in 1d and 2d. In Section 4, we propose the algorithm for the Schrödinger systems describing the surface hopping phenomenon in 1d and 2d. The numerical experiments are given in Section 5. The paper is concluded in Section 6.

2. **A system of Schrödinger equations arising from the Born-Oppenheimer approximation.**

When considering the transition of the wave functions between different potential energy levels, a system of coupled Schrödinger equations arise

in the Born-Oppenheimer approximation [3]. If we only consider the transition between two energy levels, the so-called diabatic representation of the system is of the form [1]

$$\begin{aligned} i\varepsilon \begin{pmatrix} \varphi_1(\mathbf{x},t) \\ \varphi_2(\mathbf{x},t) \end{pmatrix}_t &= -\frac{\varepsilon^2}{2}\triangle \begin{pmatrix} \varphi_1(\mathbf{x},t) \\ \varphi_2(\mathbf{x},t) \end{pmatrix} \\ &\quad + \begin{pmatrix} V_{11}(\mathbf{x}) & V_{12}(\mathbf{x}) \\ V_{21}(\mathbf{x}) & V_{22}(\mathbf{x}) \end{pmatrix} \begin{pmatrix} \varphi_1(\mathbf{x},t) \\ \varphi_2(\mathbf{x},t) \end{pmatrix}, \qquad (2) \\ \begin{pmatrix} \varphi_1(\mathbf{x},0) \\ \varphi_2(\mathbf{x},0) \end{pmatrix} &= \begin{pmatrix} \varphi_{1,0}(\mathbf{x}) \\ \varphi_{2,0}(\mathbf{x}) \end{pmatrix}, \end{aligned}$$

where $\varphi_{1,2}(\mathbf{x},t)$ are the wave functions, $V(\mathbf{x}) = (V_{ij}(\mathbf{x}))$, $i,j = 1,2$, is the symmetric potential matrix, and ε, a dimensionless small parameter, is the square root of the ratio of the mass of the electrons to that of the nuclei. To obtain the wave functions corresponding to the energy levels, one needs to diagonalize the potential matrix. Suppose $U(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}))$, where $u_1(\mathbf{x})$ and $u_2(\mathbf{x})$ are the two orthonormal eigenvectors of $V(\mathbf{x})$, then $V(\mathbf{x})$ can be diagonalized as

$$\begin{pmatrix} E_1(\mathbf{x}) & 0 \\ 0 & E_2(\mathbf{x}) \end{pmatrix} = U^{-1}(\mathbf{x}) \begin{pmatrix} V_{11}(\mathbf{x}) & V_{12}(\mathbf{x}) \\ V_{21}(\mathbf{x}) & V_{22}(\mathbf{x}) \end{pmatrix} U(\mathbf{x}).$$

Here $E_1(\mathbf{x})$ and $E_2(\mathbf{x})$ are the two potential energies corresponding to two energy levels.

Using the transformation

$$\begin{pmatrix} \varphi_1(\mathbf{x},t) \\ \varphi_2(\mathbf{x},t) \end{pmatrix} = U(\mathbf{x}) \begin{pmatrix} \psi_1(\mathbf{x},t) \\ \psi_2(\mathbf{x},t) \end{pmatrix},$$

(2) can be turned into

$$\begin{aligned} i\varepsilon \begin{pmatrix} \psi_1(\mathbf{x},t) \\ \psi_2(\mathbf{x},t) \end{pmatrix}_t &= -\frac{\varepsilon^2}{2}\triangle \begin{pmatrix} \psi_1(\mathbf{x},t) \\ \psi_2(\mathbf{x},t) \end{pmatrix} \\ &\quad + \begin{pmatrix} E_1(\mathbf{x}) & 0 \\ 0 & E_2(\mathbf{x}) \end{pmatrix} \begin{pmatrix} \psi_1(\mathbf{x},t) \\ \psi_2(\mathbf{x},t) \end{pmatrix} + D, \qquad (3) \end{aligned}$$

where D is given by

$$\begin{aligned} D &= -\frac{\varepsilon^2}{2} U^{-1}(\mathbf{x}) \nabla U(\mathbf{x}) \cdot \nabla \begin{pmatrix} \psi_1(\mathbf{x},t) \\ \psi_2(\mathbf{x},t) \end{pmatrix} \\ &\quad -\frac{\varepsilon^2}{2} U^{-1}(\mathbf{x}) \triangle U(\mathbf{x}) \begin{pmatrix} \psi_1(\mathbf{x},t) \\ \psi_2(\mathbf{x},t) \end{pmatrix}. \end{aligned}$$

Note that $\varphi_{1,2}(\mathbf{x},t)$ do not correspond to wave functions on each of the energy levels $E_{1,2}(\mathbf{x})$, but $\psi_{1,2}(\mathbf{x},t)$ do. We generally assume U, U^{-1}, $\psi_{1,2} = O(1)$, and $\nabla\psi_{1,2} = O(1/\varepsilon)$ (due to the oscillatory nature of $\psi_{1,2}$). So D is small (thus ignored) compared to the other terms in (3) as long as $\nabla U(\mathbf{x})$ and $\triangle U(\mathbf{x})$ are not large, e.g., if their sizes are $o(1/\varepsilon)$ and $o(1/\varepsilon^2)$ respectively for actually used value of ε. This is considered to be true as long as $E_{1,2}(\mathbf{x})$ are not close to each other. One way to estimate $\nabla U(\mathbf{x})$ is to estimate the change of $u_1(\mathbf{x})$ and $u_2(\mathbf{x})$ with respect to a small change of $\mathbf{x}$ [27]. An easy way to see that $U^{-1}(\mathbf{x})\nabla U(\mathbf{x})$ is

bounded for smooth $V(\mathbf{x})$ and well separated potential surfaces $E_j(\mathbf{x})$, is to notice that $U^{-1}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}))^T$ and ([5])

$$u_k^T(\mathbf{x}) \nabla u_j(\mathbf{x}) = \frac{u_k^T(\mathbf{x}) \nabla V(\mathbf{x}) u_j(\mathbf{x})}{E_j(\mathbf{x}) - E_k(\mathbf{x})}, \quad j, k = 1, 2, j \neq k.$$

In the case where D is negligible, one can obtain two decoupled Schrödinger equations and the semi-classical limits can be taken [8]. However, when the values of $E_{1,2}(\mathbf{x})$ are close, $\nabla U(\mathbf{x})$ may become large and so does D. In that case, there can be transitions of significant size between ψ_1 and ψ_2. This is the so-called non-adiabatic process. For special potential surfaces, the Landau-Zener formula [19, 43] can be used to determine the transition probability, which is the basis of many surface hopping methods. If V_{ij}, $i, j = 1, 2$, is discontinuous, then $\nabla U(\mathbf{x})$ and $\triangle U(\mathbf{x})$ become infinite and again transition generally happens at the discontinuities of V_{ij}, $i, j = 1, 2$.

In (3), when D is negligible, one can use GB for the decoupled Schrödinger equations. However, as we analyzed previously, D can be large in some situations. Then one has to go back to (2). Notice that GB can not be used for (2), since the non-diagonal terms in the potential matrix will prevent the Gaussian beams from keeping their shapes. Therefore, it is natural to seek a hybrid method consisting of a method solving (2) directly in the hopping zone and GB away from it.

3. The algorithm for the 1d/2d Schrödinger equation with discontinuous potentials.

In this section, we will propose our algorithm for the Schrödinger equation with discontinuous potentials in the 1d and 2d cases. Notice that our method can be naturally generalized for higher dimensional problems. In Subsection 3.1, we will first review the time-splitting spectral method (TSSP) and GB. In Subsection 3.2, we will propose our hybrid method.

3.1. The time-splitting spectral method and the Gaussian beam method.

3.1.1. *The time-splitting spectral method.* To solve the Schrödinger equation (1), we use the time-splitting spectral method. The algorithm is as follows.

For (1), an operator splitting method is used to obtain two equations

$$i\varepsilon\psi(\mathbf{x}, t)_t = -\frac{\varepsilon^2}{2}\triangle\psi(\mathbf{x}, t), \tag{4}$$

$$i\varepsilon\psi(\mathbf{x}, t)_t = V(\mathbf{x})\psi(\mathbf{x}, t), \tag{5}$$

which will be solved alternatively at every time step. For (4), the spectral method is used spatially and exact time integration is used; for (5), the ODE can be solved exactly. More details are referred to [2]. If the Strang splitting is used, the error caused by the splitting in time is $O\left(\Delta t_{\text{TSSP}}^2\right)$, where Δt_{TSSP} is the time step for TSSP. TSSP has the spectral accuracy in space. In practice, one needs to set Δt_{TSSP} and $\triangle x$ to be of order $O(\varepsilon)$.

Notice that TSSP is just one of the numerical methods for the Schrödinger equation. Other numerical methods can also be used here as alternatives.

3.1.2. *The Gaussian beam method.* A more efficient asymptotic method to solve the Schrödinger equation (1) is GB, which decomposes $\psi(\mathbf{x}, t)$ into Gaussian beams, i.e.

$$\psi(\mathbf{x}, t) = \sum G_k(\mathbf{x}, t), \tag{6}$$

where

$$G_k(\mathbf{x},\, t) = A_k e^{\frac{i}{\varepsilon}\left(\mathbf{p}_k\cdot(\mathbf{x}-\mathbf{y}_k)+\frac{1}{2}(\mathbf{x}-\mathbf{y}_k)^T M_k(\mathbf{x}-\mathbf{y}_k)\right)}. \tag{7}$$

Here $\mathbf{y}_k$ and $\mathbf{p}_k$ are real vectors representing the center and the momentum of the Gaussian beam. A_k is a complex number. M_k is a complex symmetric matrix whose imaginary part is positive definite. The parameters of each $G_k(x,t)$ evolve according to the ODE system [6, 9, 10, 28, 30]

$$\begin{aligned}
\frac{d\mathbf{y}_k}{dt} &= \mathbf{p}_k, \\
\frac{d\mathbf{p}_k}{dt} &= -\nabla_{\mathbf{x}} V(\mathbf{y}_k), \\
\frac{dM_k}{dt} &= -M_k^2 - \nabla_{\mathbf{x}}^2 V(\mathbf{y}_k), \\
\frac{dS_k}{dt} &= \frac{1}{2}|\mathbf{p}_k|^2 - V(\mathbf{y}_k), \\
\frac{dA_k}{dt} &= -\frac{1}{2}(Tr(M_k))\, A_k.
\end{aligned} \tag{8}$$

We are going to use GB in the Lagrangian formulation, namely, solving (8) for each k with $\mathbf{y}_k - \mathbf{y}_{k-1} = O(\sqrt{\varepsilon})$, $\forall k$, and then superimpose all the beams according to (6). The time step Δt_{GB} is set to be $O(1)$.

3.2. **The hybrid method.**

3.2.1. *Main algorithm.* We suppose that in (1) the potential $V(\mathbf{x})$ only has isolated singularities, i.e., discontinuities in the eigenvectors, in the domain. Without loss of generality, assume there is one singularity at the origin in domain C_0. For the 1d case, we assume C_0 is $[d_1,\, d_2]$ and for the 2d case $[d_1,\, d_2]\times[d_1,\, d_2]$. Around C_0, we solve the Schrödinger equation by TSSP. When the wave goes out of that area, we turn it into Gaussian beams.

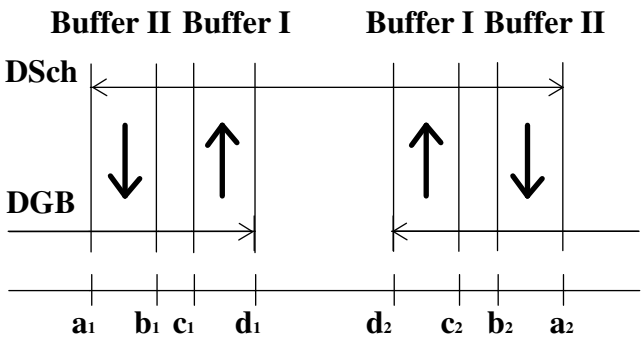

FIGURE 1. Illustration of the hybrid method in 1d.

Decompose the solution $\psi(\mathbf{x},\, t)$ into two parts

$$\psi(\mathbf{x}, t) = F(\mathbf{x}, t) + G(\mathbf{x}, t) + \text{tol}, \tag{9}$$

where 'tol' is the numerical tolerance.

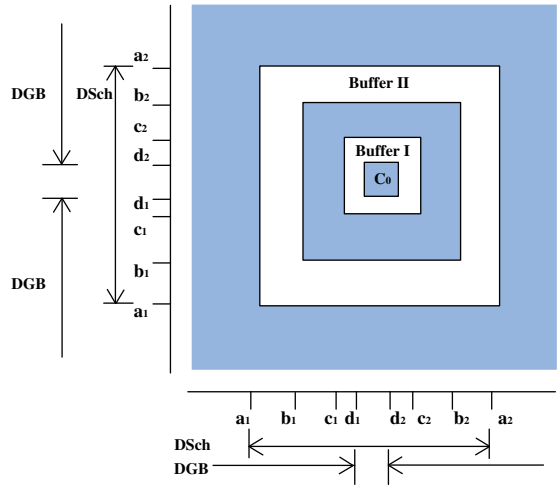

FIGURE 2. Illustration of the hybrid method in 2d.

In (9), $F(\mathbf{x}, t)$ is the part of $\psi(\mathbf{x}, t)$ that will be computed by TSSP, and has a compact support with the support set $[a_1, a_2]$ in 1d or $[a_1, a_2] \times [a_1, a_2]$ in 2d, which will be referred to as the domain of the Schrödinger solver (DSch).

$G(\mathbf{x}, t) = \sum G_k(\mathbf{x}, t)$ is the part of $\psi(\mathbf{x}, t)$ that will be computed by GB. $G_k(\mathbf{x}, t)$ are Gaussian beams, which are of the form (7). The arguments of the Gaussian beams are $\text{Arg}(G_k(\mathbf{x}, t)) = [\mathbf{y}_k, \mathbf{p}_k, A_k, S_k, M_k]$.

$G(\mathbf{x}, t)$ has the support set $(-\infty, d_1) \cup (d_2, +\infty)$ in 1d or $\Omega \backslash [d_1, d_2] \times [d_1, d_2]$ in 2d, which will be referred to as the domain of the Gaussian beam method (DGB). Rigorously speaking, every Gaussian beam has the support on the whole space. Here we abuse the term and mean the set where $1 - O(\varepsilon)$ of the total mass is in. For example, for the 1d case, using the standard normal table one can calculate that $1 - \varepsilon$ of the L_2-norm of the wave function is within $\left[y_k - \alpha(\varepsilon)\sqrt{\varepsilon/2M_k},\ y_k + \alpha(\varepsilon)\sqrt{\varepsilon/2M_k}\right]$, where $\alpha(\varepsilon) = 2$ when $\varepsilon = 0.05$ and $\alpha(\varepsilon) = 3$ when $\varepsilon = 0.003$. For the 2d case, one needs to diagonalize the covariance matrix first and then uses the standard normal table to find the support.

'tol' is the difference between $\psi(\mathbf{x}, t)$ and $F(\mathbf{x}, t) + G(\mathbf{x}, t)$, which is defined at the beginning of the computation and does not depend on the small parameter ε.

We use a mesh to solve the Schrödinger equation on DSch, i.e., $x_j = j\Delta x$ (1d) or $\mathbf{x}_{jk} = (j, k) \cdot \Delta\mathbf{x}$ (2d). For GB, we use the Lagrangian formulation so there is no mesh on DGB for $G(\mathbf{x}, t)$. We illustrate the hybridization in Figure 1 (1d) and Figure 2 (2d).

The Gaussian beams with centers in Buffer I will be turned into wave functions with pointwise-values on the mesh and added to $F(\mathbf{x}, t)$. The part of $F(\mathbf{x}, t)$ that enters Buffer II will be turned into Gaussian beams.

We discretize the time as $t_n = n\Delta t$. Notice Δt is the time step between two steps of conversions in Buffer II (Sch2GB). It is different from Δt_{TSSP} and Δt_{GB},

the time step for TSSP and that for solving the ODEs in GB introduced previously. Δt is larger than Δt_{TSSP} and Δt_{GB}. So as to reduce the number of conversion, yet it should be small enough so that no wave from the inner part of DSch will travel through Buffer II during Δt without being converted to Gaussian beams. A condition on Δt is

$$\Delta t \leq \min_{i=1,2,\,\mathbf{x}\in\Omega} \frac{|b_i - a_i|}{|\mathbf{u}(\mathbf{x})|}, \tag{10}$$

where Ω is the computational domain and the velocity $\mathbf{u}(\mathbf{x})$ is defined as

$$\mathbf{u}(\mathbf{x}) = \frac{\varepsilon \operatorname{Im}(\psi^* \nabla \psi)}{|\psi|^2}, \tag{11}$$

where ψ^* is the complex conjugate of ψ.

The main algorithm:

1. Evolution. In each time period $[t_n, t_{n+1}]$, we compute the evolutions of $F(\mathbf{x}, t)$ on DSch by TSSP, and $G(\mathbf{x}, t)$ by GB on DGB, i.e., we obtain $F(\mathbf{x}_i, t_{n+1})$ and $\operatorname{Arg}(G_k(t_{n+1}))$.
2. Schrödinger to GB. In Buffer II, apply Alg. Sch2GB.
3. GB to Schrödinger. In Buffer I, apply Alg. GB2Sch.

The algorithms Alg. Sch2GB and Alg. GB2Sch standing for the conversions between $F(\mathbf{x}, t)$ and $G(\mathbf{x}, t)$, will be explained in the next subsections.

Remark 1. The separation of Buffer I and Buffer II is necessary. If we do not do so and, instead, have only one buffer zone to convert the wave from one form to the other, then at every time step, right after $F(\mathbf{x}, t)$ is partially turned to $G(\mathbf{x}, t)$ at step 2, that part of $G(\mathbf{x}, t)$ will be added back into $F(\mathbf{x}, t)$ at step 3. At the next time step, this wasteful process will repeat. However, by separating Buffer I from Buffer II, after every conversion, the wave will not be immediately turned back into its previous form.

The widths of the buffer zones are generally $O(\sqrt{\varepsilon})$, the order of the width of a Gaussian beam, since we want to extract Gaussian beams from and put them back into those buffer zones. The second step in Alg. GB2Sch, which will be explained in the following, is to make sure the supports of the Gaussian beams being put into the buffer zones thinner than the widths of the buffer zones. Also proposed in the following, the localization step in Alg. Sch2GB and the restriction of $|\text{Im}M|$ in Alg. GBR, e.g. $|\text{Im}M| \leq 1/\sqrt{\varepsilon}$, will help to make sure the extracted Gaussian beams are not wider than the widths of the buffer zones.

Remark 2. The time step Δt here determines how often the program converts between $F(\mathbf{x}, t)$ and $G(\mathbf{x}, t)$. It has little to do with the time steps used by TSSP and GB, which are generally much smaller.

3.2.2. *Algorithm GB2Sch.* The algorithm GB2Sch is applied to $G_k(\mathbf{x}, t)$ for all k.

Algorithm GB2Sch

1. Find the Gaussian beams to be converted. For each of the Gaussian beams, $G_k(\mathbf{x}, t)$, check its parameter $\mathbf{y}_k$ and $\mathbf{p}_k$. Select the Gaussian beams whose $\mathbf{y}_k$ are in Buffer I and $\mathbf{y}_k \cdot \mathbf{p}_k < 0$ (going towards the origin).
2. Check the support of the selected Gaussian beams in the first step. If the support of $G_k(\mathbf{x}, t)$ is contained in $[b_1, b_2]$ (1d) or $[b_1, b_2] \times [b_1, b_2]$ (2d), add $G_k(\mathbf{x}, t)$ to $F(\mathbf{x}, t)$. If not, we need to decompose $G_k(\mathbf{x}, t)$ into smaller Gaussian beams and only add those whose supports are in $[b_1, b_2]$ (1d) or

$[b_1, b_2] \times [b_1, b_2]$ (2d) to $F(\mathbf{x}, t)$. There are different ways for the decomposition, e.g. [10, 29, 34, 35, 36, 42].

Remark 3. In the first step, if we do not assume the center of C_0 is the origin, but any point $\mathbf{x}_c$, we should change the condition to $(\mathbf{y}_k - \mathbf{x}_c) \cdot \mathbf{p}_k < 0$.

Remark 4. For the second step, we check the support of the Gaussian beams $G_k(\mathbf{x}, t)$ the same way as we define the support of them in Subsection 3.2.1.

3.2.3. *Algorithm Sch2GB.* We apply Alg. Sch2GB in Buffer II, i.e., $[a_1, b_1]$ and $[b_2, a_2]$ in 1d or rectangles, i.e., $[b_2, a_2] \times [a_1, a_2]$, $[a_1, b_1] \times [a_1, a_2]$, $[a_1, a_2] \times [b_2, a_2]$ and $[a_1, a_2] \times [a_1, b_1]$ in 2d. For the sake of simplicity, in the following we will describe the algorithm on $[b_2, a_2]$ or $[b_2, a_2] \times [a_1, a_2]$.

The following constants are defined at the beginning. They will be used in the algorithm.

x_{check}: $b_2 < x_{\text{check}} < a_2$.

V_{thrhd}: A threshold for the probability density defined at the beginning of the program.

E_{thrhd}: A threshold for the probability density defined at the beginning of the program, a small number.

η: A small percentage number.

Algorithm Sch2GB:

1. Determine whether to start the conversion.
   - In the 1d case, on $[b_2, a_2]$ check if $|F(x_{\text{check}}, t)|^2 > V_{\text{thrhd}}$. Or in the 2d case, on $[b_2, a_2] \times [a_1, a_2]$ check if $\max_{x_2} \left\{ |F((x_{\text{check}}, x_2), t)|^2 \right\} > V_{\text{thrhd}}$.
     - If it is not true, then this algorithm ends and go back to the main algorithm.
     - If it is true, then continue to step 2.
2. Localize the function in Buffer II and obtain the function to be converted.
   - $F_L(\mathbf{x}, t) = F(\mathbf{x}, t) \cdot L(\mathbf{x})$, where $L(\mathbf{x})$ is a smooth damping function with a support set on $[b_2, a_2]$ or $[b_2, a_2] \times [a_1, a_2]$.
   - $F_{\text{rest}}(\mathbf{x}, t) = F(\mathbf{x}, t) - F_L(\mathbf{x}, t)$;
3. Extract Gaussian beams from $F_L(\mathbf{x}, t)$.
   - Tot $= \int |F_L(\mathbf{x}, t)|^2 d\mathbf{x}$;
   - $k = 0$;
   - While the probability density of $\int |F_L(\mathbf{x}, t)|^2 d\mathbf{x} > E_{\text{thrhd}}$ and $\int |F_L(\mathbf{x}, t)|^2 d\mathbf{x} > \eta \cdot$ Tot, repeat the following.
     - $k = k + 1$;
     - Invoke Alg. GBR to extract a Gaussian beam $G_k(\mathbf{x}, t)$ from $F_L(\mathbf{x}, t)$.
     - $F_{\text{temp}}(\mathbf{x}, t) = F_L(\mathbf{x}, t) - G_k(\mathbf{x}, t)$;
     - $F_L(\mathbf{x}, t) = F_{\text{temp}}(\mathbf{x}, t) \cdot L(\mathbf{x})$; (Localizing)
     - $F_{\text{rest}}(\mathbf{x}, t) = F_{\text{rest}}(\mathbf{x}, t) + F_{\text{temp}}(\mathbf{x}, t) - F_L(\mathbf{x}, t)$;
4. Add the remaining function back.
   - $F(\mathbf{x}, t) = F(\mathbf{x}, t) + F_{\text{rest}}(\mathbf{x}, t)$

In the above, Alg. GBR is the algorithm to turn pointwise-valued wave functions into Gaussian beams. It will be explained in the next subsection.

Remark 5. Step 1 is necessary for improving the performance, since Alg. GBR is costly and we do not want to use it for every time step. We use it only when the wave amplitude reaches the threshold V_{thrhd} at x_{check}. In the numerical examples we

used $x_{\text{check}} = \frac{a_2+3b_2}{4}$, $\frac{a_2+b_2}{2}$ and $\frac{5a_2+3b_2}{8}$. It shows that the algorithm is not sensitive to the position of x_{check} as long as it is not too close to a_2. A too large V_{thrhd} may lead to an error since waves with small amplitudes may not trigger the program. However, too small V_{thrhd} will cause the program to be invoked for unnecessarily too many times and too many Gaussian beams to be produced.

Remark 6. The use of the smooth damping function $L(\mathbf{x})$ in steps 3 and 4 is important. Because if one, instead, uses $F_L(\mathbf{x}, t) = F(\mathbf{x}, t)\chi_{B_{II}}(\mathbf{x})$, where $\chi_{B_{II}}(\mathbf{x}) = 1$ in Buffer II and $\chi_{B_{II}}(\mathbf{x}) = 0$ elsewhere, the amplitude of $F_L(\mathbf{x}, t)$ becomes discontinuous at the inner boundary of Buffer II. (There is no discontinuity at the outer boundary of Buffer II, since the wave is coming from C_0 and it is always zero at the outer boundary.) As a consequence, Alg. GBR, explained below, will produce many Gaussian beams, thus become very inefficient. The 1d localizing function $L(x)$ on $[a, b]$ is defined as

$$L(x) = \hat{L}\left(\frac{x}{b-a}\right), \quad a \le x \le b,$$

where

$$\hat{L}(x) = \left(\frac{\tanh(20(x-q))+1}{2}\right)\left(\frac{\tanh(-20(x-(1-q)))+1}{2}\right), \quad 0 \le x \le 1, \tag{12}$$

with $0 < q < 0.5$.

For the 2d case, we use a similar one on $[a, b] \times [c, d]$

$$L(x_1, x_2) = \hat{L}\left(\frac{x_1}{b-a}, \frac{x_2}{d-c}\right), \quad (x_1, x_2) \in [a, b] \times [c, d],$$

where

$$\hat{L}(x_1, x_2) = \prod_{k=1,2}\left(\frac{\tanh(20(x_k-q))+1}{2}\right)\left(\frac{\tanh(-20(x_k-(1-q)))+1}{2}\right), \quad 0 \le x_{1,2} \le 1. \tag{13}$$

Remark 7. At Step 3, we check $\int |F_L(\mathbf{x}, t)|^2 d\mathbf{x} > E_{\text{thrhd}}$ and $\int |F_L(\mathbf{x}, t)|^2 d\mathbf{x} > \eta \cdot \text{Tot}$ so that the program stops if the part of wave that has not been turned into Gaussian beams is relatively and absolutely small. Similar to V_{thrhd}, a too large E_{thrhd} and η will cause an error larger than tolerance. A too small E_{thrhd} and η will cause Alg. GBR to be invoked for too many times which will produce too many Gaussian beams with small amplitudes.

3.2.4. *Algorithm GBR.* Alg. GBR, standing for Gaussian beam recovering, extracts the parameters of Gaussian beams from an oscillatory function, $F(\mathbf{x})$, on a discrete mesh. We use the one in [37] with minor modifications. We briefly review the algorithm we use here. This is the version for both 1d and 2d problems.

Define the energy function and the energy norm as

$$E(u(\mathbf{x})) = (V(\mathbf{x}) - V_{\min})|u(\mathbf{x})|^2 + \frac{1}{2}|\nabla u(\mathbf{x})|^2, \tag{14}$$

and

$$\|u(\mathbf{x})\|_E^2 = \int (V(\mathbf{x}) - V_{\min})|u(\mathbf{x})|^2 + \frac{1}{2}|\nabla u(\mathbf{x})|^2 d\mathbf{x}, \tag{15}$$

correspondingly, where $V_{\min} = \min_{\mathbf{x}} \{V(\mathbf{x})\}$. The related inner product is

$$\langle F(\mathbf{x}), G(\mathbf{x})\rangle_E = \int (V(\mathbf{x}) - V_{\min}) F(\mathbf{x}) G(\mathbf{x})^* + \frac{1}{2}\nabla F(\mathbf{x}) \cdot \nabla G(\mathbf{x})^* \, d\mathbf{x}.$$

Algorithm GBR:

1. Estimate Gaussian beam parameters
   - Estimate the Gaussian beam center
     - Let $\tilde{\mathbf{y}} = \arg\max_{\mathbf{y}} \{E(u(\mathbf{y}))\}$.
   - Estimate the propagation direction
     - Let $G(\mathbf{x}) = e^{-\frac{1}{2}k|\mathbf{x}-\mathbf{y}|^2}$
     - Let $\tilde{\mathbf{p}} = \arg\max_{\mathbf{p}} \{|\mathcal{F}[u(\mathbf{x}) G(\mathbf{x})](\mathbf{p})|\}$ with the restriction $\sqrt{\varepsilon} \le |\tilde{\mathbf{p}}| \le \sqrt{1/\varepsilon}$, where $\mathcal{F}$ is the Fourier transform.
   - Estimate $\mathrm{Im}M$
     - Let $\left[\mathrm{Im}\tilde{M}, \tilde{a}\right] = \arg\max_{\mathrm{Im}M, a} \left\| a e^{-\frac{1}{2\varepsilon}\mathrm{Im}M|\mathbf{x}-\tilde{\mathbf{y}}|^2/2} - |u(\mathbf{x})| \right\|_E$ with $\mathrm{Im}M = I$, the identity matrix, and $a = |u(\mathbf{x})|$ as the initial parameters and with the restrictions:
       - $\mathrm{Im}\tilde{M}$ is symmetric and positive definite
       - $\left|\mathrm{Im}\tilde{M}\right| \le 1/\sqrt{\varepsilon}$
   - Estimate $\mathrm{Re}M$
     - $\mathrm{Re}\tilde{M} = \arg\max_{\mathrm{Re}M} \left\| u - \frac{\langle u, \tilde{G}\rangle_E}{\|\tilde{G}\|_E^2}\tilde{G} \right\|_E$ with $\mathrm{Re}M = I$ as the initial parameter and with the restriction that $\mathrm{Re}\tilde{M}$ is symmetric
2. Minimize the difference between the Gaussian beam and $u(\mathbf{x})$ in the energy norm using $\tilde{\mathbf{y}}$, $\tilde{\mathbf{p}}$, and $\tilde{M}$ as the initial Gaussian beam parameters.

- Let $\tilde{G}(\mathbf{x}, t) = e^{\frac{i}{\varepsilon}\left(\tilde{\mathbf{p}}\cdot(\mathbf{x}-\tilde{\mathbf{y}}) + \frac{1}{2}(\mathbf{x}-\tilde{\mathbf{y}})^T \tilde{M}(\mathbf{x}-\tilde{\mathbf{y}})\right)}$
- Let $[\mathbf{y}, \mathbf{p}, M] = \arg\max_{\mathbf{y},\mathbf{p},M} \left\| u - \frac{\langle u, \tilde{G}\rangle_E}{\|\tilde{G}\|_E^2}\tilde{G} \right\|_E$ with restrictions:
  - $\mathrm{Im}\tilde{M}$ is symmetric and positive definite
  - $\left|\mathrm{Im}\tilde{M}\right| \le 1/\sqrt{\varepsilon}$
- Let $G(\mathbf{x}, t) = e^{\frac{i}{\varepsilon}\left(\mathbf{p}\cdot(\mathbf{x}-\mathbf{y}) + \frac{1}{2}(\mathbf{x}-\mathbf{y})^T \tilde{M}(\mathbf{x}-\mathbf{y})\right)}$
- Let $A = \frac{\langle u, G\rangle_E}{\|G\|_E^2}$

Remark 8. Comparing to the original algorithm in [37], we have two more steps in step 1, which are to estimate the imaginary and real parts of M. In [37], they use $\mathrm{Im}M = I_d$ and $\mathrm{Re}M = 0$ as the initial guesses for the next nonlinear minimization step.

Notice that $\mathrm{Im}M$ and $|A|$ determine the profile of the Gaussian beam. The additional step for $\mathrm{Im}M$ will give a better initial guess of $\mathrm{Im}M$ for the later nonlinear minimization. In practice, we find the step to estimate $\mathrm{Re}M$ also helpful to reduce the number of Gaussian beams. They are both nonlinear minimizations for fewer parameters, so they converge faster. More details about the nonlinear minimization with constraints are discussed in [37].

We only use the additional steps for the 2d case, since in the 1d case, the original algorithm in [37] is already very good. A simple comparison of the numbers of the Gaussian beams produced by the two versions of the algorithm is in Section 5.1.

4. **The algorithm for 1d/2d system of Schrödinger equations for surface hopping.** For the system of the Schrödinger equations (2), we can not directly use GB, since the coupling terms will prevent the Gaussian beams from maintaining their Gaussian shapes.

In (3), when D is small, one can ignore it, then the remaining part of (3) is a decoupled system of equations, which can be used to approximate (2). (This is the essence of the Born-Oppenheimer approximation [3].) We then apply the Gaussian beam method for $\psi_{1,2}$, respectively.

We will assume that D is large only around the origin, namely in C_0. Then one can use the algorithm in the previous section with some modifications.

We write the solutions to the systems (2) and (3) into

$$\begin{aligned} \varphi_1(\mathbf{x}, t) &= F_1(\mathbf{x}, t) + \tilde{G}_1(\mathbf{x}, t) + \mathrm{tol}_1, \\ \varphi_2(\mathbf{x}, t) &= F_2(\mathbf{x}, t) + \tilde{G}_2(\mathbf{x}, t) + \mathrm{tol}_2, \end{aligned}$$

and

$$\begin{aligned} \psi_1(\mathbf{x}, t) &= \tilde{F}_1(\mathbf{x}, t) + G_1(\mathbf{x}, t) + \mathrm{tol}_3, \\ \psi_2(\mathbf{x}, t) &= \tilde{F}_2(\mathbf{x}, t) + G_2(\mathbf{x}, t) + \mathrm{tol}_4, \end{aligned}$$

where

$$\begin{pmatrix} F_1(\mathbf{x}, t) \\ F_2(\mathbf{x}, t) \end{pmatrix} = U \begin{pmatrix} \tilde{F}_1(\mathbf{x}, t) \\ \tilde{F}_2(\mathbf{x}, t) \end{pmatrix},$$

and

$$\begin{pmatrix} \tilde{G}_1(\mathbf{x}, t) \\ \tilde{G}_2(\mathbf{x}, t) \end{pmatrix} = U \begin{pmatrix} G_1(\mathbf{x}, t) \\ G_2(\mathbf{x}, t) \end{pmatrix}.$$

We will calculate $F_{1,2}$ using TSSP on DSch, and $G_{1,2}$ using GB on DGB. In Buffer I, we need to add one step, i.e., turn $\{G_{1,k}, G_{2,k}\}$ into $\left\{\tilde{G}_{1,k}, \tilde{G}_{2,k}\right\}$ and then add the latter ones to $F_{1,2}$ using an algorithm similar to Alg. GB2Sch. Correspondingly, in Buffer II, we do the checking (Alg. Sch2GB step 1) for $\{F_1, F_2\}$. If one of them triggers the process, then we turn them into $\left\{\tilde{F}_1, \tilde{F}_2\right\}$ and extract Gaussian beams from $\left\{\tilde{F}_1, \tilde{F}_2\right\}$.

Remark 9. One might extend the method in [36] to surface hopping problems. One can see that there is no essential difference in transforming the systems of equations from one to the other and dividing the whole space according to whether the diagonalized system is a good approximation to the original one. The main difference between our method and theirs is that we use a fixed mesh around the hopping area of the system instead of local meshes moving with the wave. The advantage of their method using a moving mesh is that, when the wave is away from the hopping area, it evolves as Gaussian beams and the computational cost is cheap.

However, there are two obvious situations when their method does not perform well. First, when many Gaussian beams are entering the hopping area, one needs to make local meshes for each of the Gaussian beams and evolve each of them separately. Usually, for an initial wave of width $O(1)$, one needs to decompose it into $O(1/\sqrt{\varepsilon})$ (1d) or $O(1/\varepsilon)$ (2d) Gaussian beams. Then one may need to solve

$O(1/\sqrt{\varepsilon})$ or $O(1/\varepsilon)$ systems of equations later. Even if at the beginning there is only one Gaussian beam, it may split into many after passing through the hopping area. Second, it is relatively hard to estimate the time when the wave in the hopping area will go out, especially when the potential in the hopping area is complicated. Third, because one converts all the wave from the local mesh to Gaussian beams simultaneously, one needs to wait until all the waves go out of the hopping area. In some situations, the domain of the local mesh can be very large, e.g., part of the wave stays in the hopping area for a long time while other parts go away very soon at a fast speed.

Our method with a fixed mesh does not have the previous drawbacks. For the first issue, when the wave in the form of the Gaussian beams enters the hopping area, they automatically merge and we only need to use TSSP on one mesh. One does not worry about the second and the third issues either, since the wave will turn from one form to the other automatically at the boundary of the hopping area.

Unlike the method in [36], our method uses TSSP in the hopping area all the time even when there is no wave in that area. Nevertheless, the computational cost of our method is fixed. For example, for the surface hopping problems where the eigenvalues of the potential matrix are close to each other at one point, we use a rectangle with width of $O(\sqrt{\varepsilon})$ to enclose it. This will result in $O\left(\varepsilon^{-d/2}\right)$ grid points, which is of the same order of the number of the Gaussian beams obtained after the initial value decomposition for waves with width $O(1)$. The time step for TSSP is $O(\varepsilon)$ and it is $O(1)$ for GB. Finally, our method is cheaper than solely using TSSP on the whole domain, yet able to capture the hopping phenomena while the original GB can not.

The error of the hybrid method, in terms of ε, are from four places: the Schrödinger solver for continuous potential ($O\left((\Delta x/\varepsilon)^m/\Delta t + \Delta t/\varepsilon\right)$ for TSSP with first order time splitting, m-order smoothness of the wave function and $\Delta x = \Delta t = O(\varepsilon)$, refer to [2]), the Gaussian beam method ($O(\sqrt{\varepsilon})$), ignoring D in (3) and Alg. Sch2GB. The term D in (3) is considered as $O(\varepsilon/d)$, where d is the size of the energy gap. In our numerical experiments, we only ignore D where d is larger than $O(\sqrt{\varepsilon})$ for fixed ε, i.e., away from the hopping area. Thus, the error by ignoring D is $O(\sqrt{\varepsilon})$. For Alg. Sch2GB, the error is due to that some of the wave will leak through the buffer zone without triggering Alg. GBR. It can be made smaller by reducing V_{thrhd}, E_{thrhd} and η in Alg. Sch3GB, which, however, may increase the computational cost as discussed in Remarks 5 and 7.

5. Numerical experiments.

5.1. Comparison of the two versions of Algorithm GBR.

In Section 3.2.3 we proposed our modified Alg. GBR. In a 2d example for comparison with the original Alg. GBR by [37] (denoted by TET), we set the domain to be $[-2, 2] \times [-2, 2]$, and $\Delta x_1 = \Delta x_2 = 1/75$.

We test the algorithms for three different functions to be turned into Gaussian beams. We denote the Gaussian beams with $[\mathbf{y}_k, \mathbf{p}_k, A_k, S_k, M_k]$ as parameters by G_k, and set $\mathbf{y}_1 = [-0.6, 0.1]$, $\mathbf{y}_2 = [-0.4, 0.1]$, $\mathbf{y}_3 = [-0.8, 0.1]$, $\mathbf{p}_k = [1.5, 1]$, $A_k = i$, $S_k = 0$, and $M_k = \begin{pmatrix} 1+2i & 0 \\ 0 & i \end{pmatrix}$, $k = 1, 2, 3$.

In case 1, $u(\mathbf{x}) = G_1(\mathbf{x})$; in case 2, $u(\mathbf{x}) = G_1(\mathbf{x}) + G_2(\mathbf{x})$; in case 3, $u(\mathbf{x}) = G_1(\mathbf{x}) + G_2(\mathbf{x}) + G_3(\mathbf{x})$.

We use "fminsearch()" in Matlab for the nonlinear minimization where we set the maximum number of iteration as 100. In the enery function and the energy norm defined in (14) and (15), we take $V(\mathbf{x}) = 0$. The program stops when 99% of the energy norm of $u(\mathbf{x})$ is turned into Gaussian beams.

The comparison of the two algorithms in terms of the time cost and the number of Gaussian beams produced is in Table 1. From the table one can see that our modified algorithm generates less Gaussian beams while the time costs are comparable to TET in those cases. However, since it is hard to analyze any nonlinear minimization process and there are many other factors involved, we can not conclude which algorithm is definitely better than the other in general.

	time cost		number of GBs	
	TET	modified	TET	modified
case 1	37s	23s	4	1
case 2	113s	103s	11	5
case 3	109s	130s	12	7

TABLE 1. Comparison of the original and modified algorithms.

5.2. **Example 1: The Schrödinger equation with a potential barrier.** The first example to test our hybrid method is to solve (1) with a quantum barrier

$$V(x) = \begin{cases} x^2, & x < -0.005 \text{ or } x > 0.005, \\ x^2 + 1, & -0.005 < x < 0.005, \end{cases}$$

with the initial value $\psi_0(x) = A_0 e^{\frac{i}{\varepsilon}\left(S_0 + p_0(x-y_0) + \frac{1}{2}M_0 x^2\right)}$, where $[y_0, p_0, A_0, S_0, M_0] = [-0.2, 1, 1, 0, 10i]$.

We set $\varepsilon = 1/200$. For the hybrid method, DSch is set to be $[-0.8, 0.8]$, $\Delta t = 0.01$, $\Delta t_{\text{TSSP}} = 1/20000$, and $\Delta x = 1/5120$. Δt_{GB} is chosen by "ode45()" in Matlab. We use the result by TSSP with $\Delta t_{\text{TSSP}} = 1/20000$ and $\Delta x = 1/5120$ for the whole domain $[-1.6, 1.6]$ as the reference. The small time and space steps are due to the discontinuity in the potential. We set

$$[a_1, b_1, c_1, d_1, d_2, c_2, b_2, a_2] = [-0.8, -0.4, -0.3, -0.1, 0.1, 0.3, 0.4, 0.8]$$

and $x_{\text{check}} = 0.5$. q in (12) is set to be 0.2.

The results using the hybrid method and TSSP are shown in Fig. 3. The first picture is the initial wave which will move to the right. The second picture shows that when the wave reaches the potential barrier, part of it transmits and the other part is reflected. The third picture is drawn at $t = 0.8$, when the transmitted and reflected waves are at the boundary of DSch and they are being turned into Gaussian beams. However, the summation of $F(x, t)$ and $G(x, t)$ is always a good approximation of $\psi(x, t)$. In the fourth picture, the two waves return because they can not go over the quadratic potential. In the fifth picture, the two waves go back toward the potential barrier at the same time and they are turned back to pointwise valued functions. In the last picture, after the two waves hit the potential barrier, they separate and move outward again.

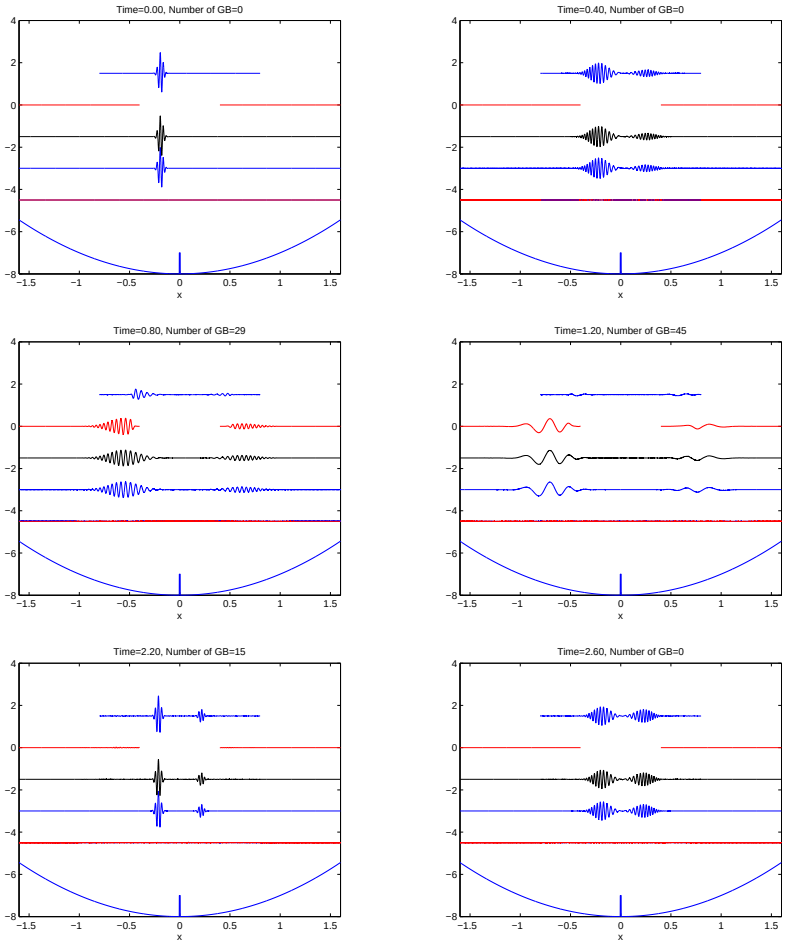

FIGURE 3. Example 1: The results using the hybrid method and TSSP. For each function, we only draw its real part. The imaginary part looks similar. In each picture, the upper most one is the result by TSSP on DSch, i.e. $F(x, t)$. The second one is $G(x, t)$. The next is $F(x, t) + G(x, t)$. The fourth is the solution by TSSP. We consider it as the reference solution, $\psi(x, t)$. The fifth is the difference between $\psi(x, t)$ and $(F(x, t) + G(x, t))$. The curve on the bottom shows the shape of the potential.

Define the relative error as

$$\text{error}(t) = \frac{\|\psi(\mathbf{x}, t) - (F(\mathbf{x}, t) + G(\mathbf{x}, t))\|_2}{\|\psi(\mathbf{x}, 0)\|_2}.$$

To demonstrate how the factors, V_{thrhd}, E_{thrhd}, and η, in Alg. Sch2GB affect the performance of the computation, we choose three different sets of numbers (Table 2) and compare the relative errors and the numbers of Gaussian beams generated.

	V_{thrhd}	E_{thrhd}	η
case 1	0.1	0.01	0.01
case 2	0.075	0.0075	0.0075
case 3	0.05	0.005	0.005

TABLE 2. Different control parameters, V_{thrhd}, E_{thrhd}, and η for comparison.

Fig. 4 is the relative error of the results and the total number of Gaussian beams for the hybrid method with different V_{thrhd}, E_{thrhd}, and η as given in Table 2. Notice that around $t = 2.6$, when almost all the Gaussian beams have entered Buffer I, the total number of Gaussian beams is very small. We can see the trend from Fig. 4 that smaller values of V_{thrhd}, E_{thrhd}, and η generally result in smaller errors but larger numbers of Gaussian beams. In practice, we need to find a good balance of them.

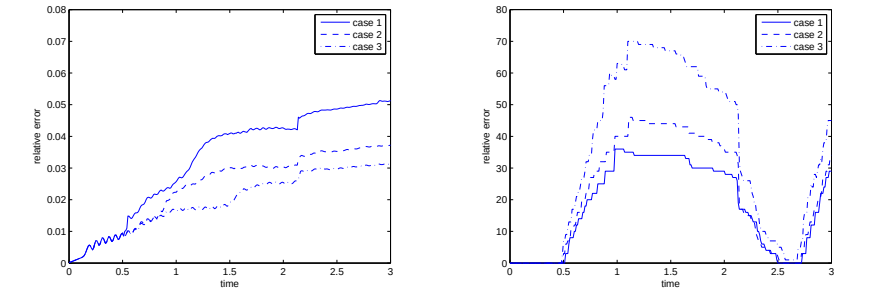

FIGURE 4. Example 1: The relative errors (left) and the numbers of Gaussian Beams (right) of the hybrid method for different control parameters in Table 2.

We can also test the stability of the algorithm with respect to the initial value. We use

$$\psi_0^\delta(x) = A_0 e^{\frac{i}{\varepsilon}\left(S_0 + p_0(x-y_0) + \frac{1}{2}M_0 x^2\right)} + \delta \sin(x)\, e^{-\frac{i}{2\varepsilon}M_0(x-0.01)^2}$$

with parameters $[y_0, p_0, A_0, S_0, M_0] = [-0.2, 1, 1, 0, 10i]$ and let $\delta = 1, 1/2, 1/4 \cdots 1/32$. The error, defined as

$$\text{error}_{l_2}^\delta(t) = \frac{\left\|\psi^\delta(\mathbf{x}, t) - \psi^0(\mathbf{x}, t)\right\|_2}{\left\|\psi^0(\mathbf{x}, 0)\right\|_2}$$

is in Table 3. We can see the errors decay with decaying δ. Other numerical experiments also suggest the stability of the algorithm with respect to the initial value.

δ	1	1/2	1/4
$\text{error}_{l_2}^\delta(2)$	1.9×10^{-1}	9.7×10^{-2}	5.3×10^{-2}
δ	1/8	1/16	1/32
$\text{error}_{l_2}^\delta(2)$	4.9×10^{-2}	2.3×10^{-2}	1.4×10^{-2}

TABLE 3. $\text{error}_{l_2}^\delta(2)$ for different δ.

5.3. **Example 2: The Schrödinger system - 1d surface hopping problem 1.** We now solve the system (2) with

$$V(x) = \begin{pmatrix} |x| & \delta \\ \delta & -|x| \end{pmatrix},$$

where $\delta = 0.05$. The initial value is

$$\begin{pmatrix} \varphi_{1,0}(x) \\ \varphi_{2,0}(x) \end{pmatrix} = U(x) \begin{pmatrix} \psi_{1,0}(x) \\ \psi_{2,0}(x) \end{pmatrix}$$

with $\psi_{1,0}(x) = A_0 e^{\frac{i}{\varepsilon}\left(S_0 + p_0(x-y_0) + \frac{1}{2}M_0 x^2\right)}$, where

$$[y_0,\, p_0,\, A_0,\, S_0,\, M_0] = [-1,\, 0.8,\, 0.3,\, 0,\, 4i],$$

and $\psi_{2,0}(x) = 0$.

We set $\varepsilon = 1/100$. For the hybrid method, DSch is set to be $[-1.4,\, 1.4]$, $\Delta t = 0.01$, $\Delta t_{\text{TSSP}} = 0.001$, and $\Delta x = 1/2048$. Δt_{GB} is chosen by "ode45()" in Matlab. We use the result by TSSP with $\Delta t_{\text{TSSP}} = 0.001$ and $\Delta x = 1/2048$ for the whole domain $[-2,\, 2]$ as the reference, and set

$$[a_1,\, b_1,\, c_1,\, d_1,\, d_2,\, c_2,\, b_2,\, a_2] = [-1.4,\, -1,\, -0.8,\, -0.6,\, 0.6,\, 0.8,\, 1,\, 1.4],$$

$x_{\text{check}} = 1.2$ and $V_{\text{thrhd}} = E_{\text{thrhd}} = \eta = 0.01$. q in (12) is set to be 0.2.

The results are shown in Fig.5. Now there are two sets of curves for $\varphi_{1,2}$, respectively. The first picture is the initial wave which will move to the right. The second picture in the first row shows that after the wave passes the hopping area, where the energy gap is smallest, part of it hops to the other energy level. The third and fourth pictures show that the waves on the two energy levels are turned into Gaussian beams and after that the upper one is turned back into the pointwise-valued function since it is reflected. The second hopping phenomenon can be seen in the last two pictures and our hybrid method obtains a good approximation to the reference solution.

The relative errors defined by

$$\text{error}_{1,2}(t) = \frac{\left\| \psi_{1,2}(\mathbf{x},\, t) - \left(\tilde{F}_{1,2}(\mathbf{x},\, t) + G_{1,2}(\mathbf{x},\, t) \right) \right\|_2}{\|\psi_1(\mathbf{x},\, 0)\|_2 + \|\psi_2(\mathbf{x},\, 0)\|_2} \tag{16}$$

are shown in Fig. 6.

Since the potential has no discontinuity and the solutions on the two levels are very close to Gaussian beams, one can see in Fig.6 that the numbers of Gaussian beams produced are small.

5.4. **Example 3: Schrödinger equation system - 1d surface hopping problem 2.** We now solve the system (2) with

$$V(x) = \begin{pmatrix} x^2 & \delta \\ \delta & -x^2 \end{pmatrix},$$

where $\delta = 0.01$.

The initial value is

$$\begin{pmatrix} \varphi_{1,0}(x) \\ \varphi_{2,0}(x) \end{pmatrix} = U(x) \begin{pmatrix} \psi_{1,0}(x) \\ \psi_{2,0}(x) \end{pmatrix}$$

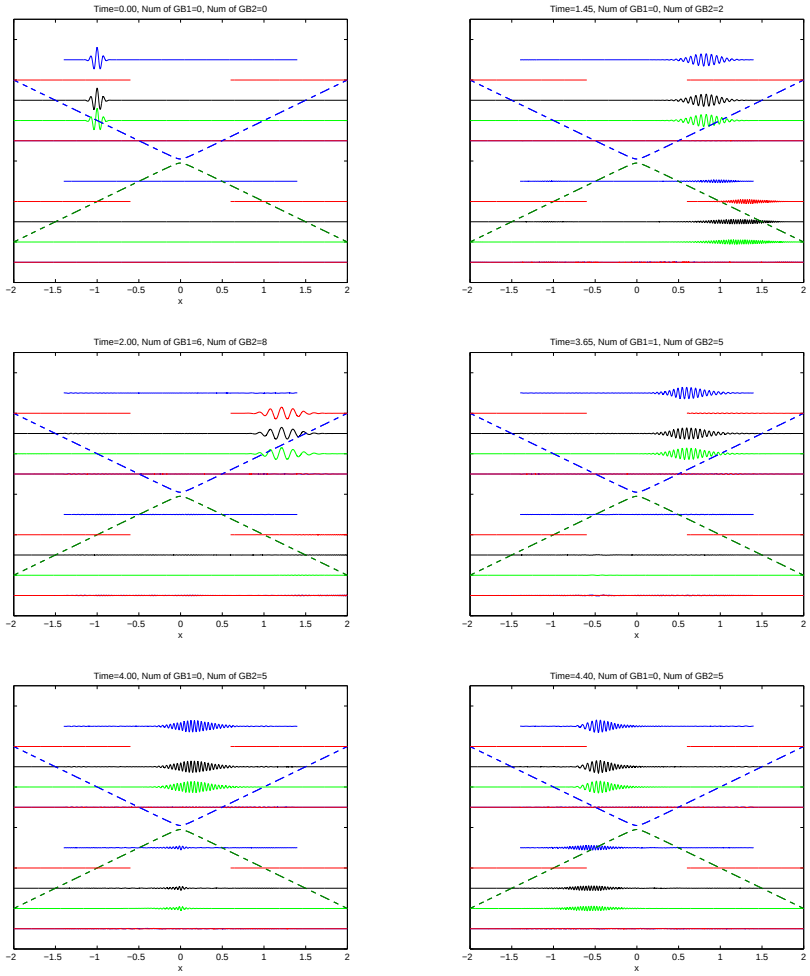

FIGURE 5. Example 2: The results using the hybrid method and TSSP. For each function, we only draw its real part. The imaginary part looks similar. In every picture, there are two solutions $\psi_{1,2}$ represented by two sets of curves with ψ_1 on the upper half and ψ_2 on the lower half. In each set of curves, the upper most one is by TSSP on DSch, i.e. $F_{1,2}(x,\, t)$. The next is $\tilde{G}_{1,2}(x,\, t)$. Then $F_{1,2}(x,\, t) + \tilde{G}_{1,2}(x,\, t)$. The fourth one is the solution by TSSP. We consider it as the reference solution, $\psi_{1,2}(x,\, t)$. The fifth is the difference between $\psi_{1,2}(x,\, t)$ and $\left(F_{1,2}(x,\, t) + \tilde{G}_{1,2}(x,\, t)\right)$. The dashed curves on the background demonstrate the the shape of the potential energies.

with $\psi_{1,0}(x) = A_0 e^{\frac{i}{\varepsilon}\left(S_0 + p_0(x-y_0) + \frac{1}{2}M_0 x^2\right)}$, where

$$[y_0,\, p_0,\, A_0,\, S_0,\, M_0] = [-1,\, 0.8,\, 0.3,\, 0,\, 4i]\,,$$

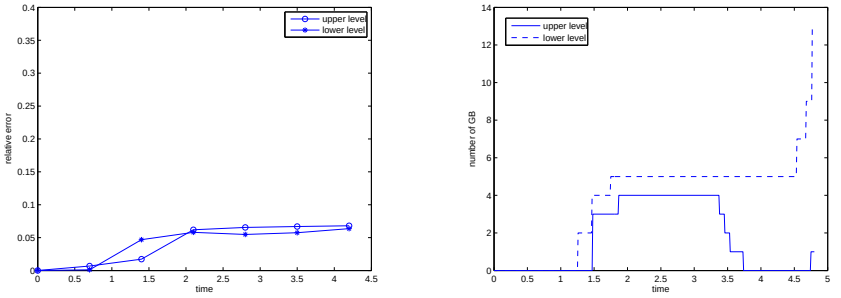

FIGURE 6. Example 2: The relative error, $\text{error}_1(t) + \text{error}_2(t)$ (left) and the numbers of Gaussian Beams (right) of the hybrid method.

and $\psi_{2,0}(x) = 0$. We set $\varepsilon = 1/100$. For the hybrid method, DSch is set to be $[-1.4, 1.4]$, $\Delta t = 0.01$, $\Delta t_{\text{TSSP}} = 0.001$, and $\Delta x = 1/1024$. Δt_{GB} is chosen by "ode45()" in Matlab. We use the result by TSSP with $\Delta t_{\text{TSSP}} = 0.001$ and $\Delta x = 1/1024$ for the whole domain $[-2, 2]$ as the reference, and set

$$[a_1, b_1, c_1, d_1, d_2, c_2, b_2, a_2] = [-1.4, -1, -0.8, -0.6, 0.6, 0.8, 1, 1.4],$$

$x_{\text{check}} = 1.2$ and $V_{\text{thrhd}} = E_{\text{thrhd}} = \eta = 0.01$. q in (12) is set to be 0.2.

The results are shown in Fig. 7. The scenario is very similar to the last numerical example. One can see two hoppings in the pictures.

The relative errors, defined the same way as in (16), and the numbers of Gaussian beams produced, are shown in Fig. 8.

5.5. **Example 4: Schrödinger equation system - 2d surface hopping problem.** We now solve system (2) in 2d with

$$V(\mathbf{x}) = \begin{pmatrix} x^2 + y^2 & \delta \\ \delta & -(x^2 + y^2) \end{pmatrix}, \quad \mathbf{x} = (x, y) \in \mathbb{R}^2, \tag{17}$$

where $\delta = 0.05$.

The initial value is

$$\begin{pmatrix} \varphi_{1,0}(\mathbf{x}) \\ \varphi_{2,0}(\mathbf{x}) \end{pmatrix} = U(\mathbf{x}, t) \begin{pmatrix} \psi_{1,0}(\mathbf{x}) \\ \psi_{2,0}(\mathbf{x}) \end{pmatrix}$$

with $\psi_{1,0}(\mathbf{x}) = A_0 e^{\frac{i}{\varepsilon}\left(S_0 + \mathbf{p_0}\cdot(\mathbf{x}-\mathbf{y_0}) + \frac{1}{2}(\mathbf{x}-\mathbf{y_0})^T M(\mathbf{x}-\mathbf{y_0})\right)}$, where $\mathbf{y_0} = [-0.5, 0.08]$, $\mathbf{p_0} = [1.5, 0]$, $A_0 = i$, $S_0 = 0$, and $M_0 = \begin{pmatrix} i & 0 \\ 0 & i \end{pmatrix}$, $\varepsilon = 1/50$, DSch $[-1.4, 1.4] \times [-1.4, 1.4]$, $\Delta t = 0.01$ and $\Delta \mathbf{x} = [1/75, 1/75]$. We use the result by TSSP with $\Delta t_{\text{TSSP}} = 0.001$ and $\Delta \mathbf{x} = [1/75, 1/75]$ for the whole domain $[-2, 2]$ as the reference. We set

$$[a_1, b_1, c_1, d_1, d_2, c_2, b_2, a_2] = [-1.4, -1, -0.8, -0.6, 0.6, 0.8, 1, 1.4],$$

$x_{\text{check}} = 1.25$, $V_{\text{thrhd}} = 0.005$, $E_{\text{thrhd}} = 0.02$, and $\eta = 0.02$. q in (13) is set to be 0.2.

The initial value and numerical results are shown in Fig. 9 to Fig. 11. The scenario is similar to the 1d numerical examples. There are two hoppings.

The relative errors, defined the same way as in (16), and the numbers of Gaussian beams produced, are shown in Fig. 12.

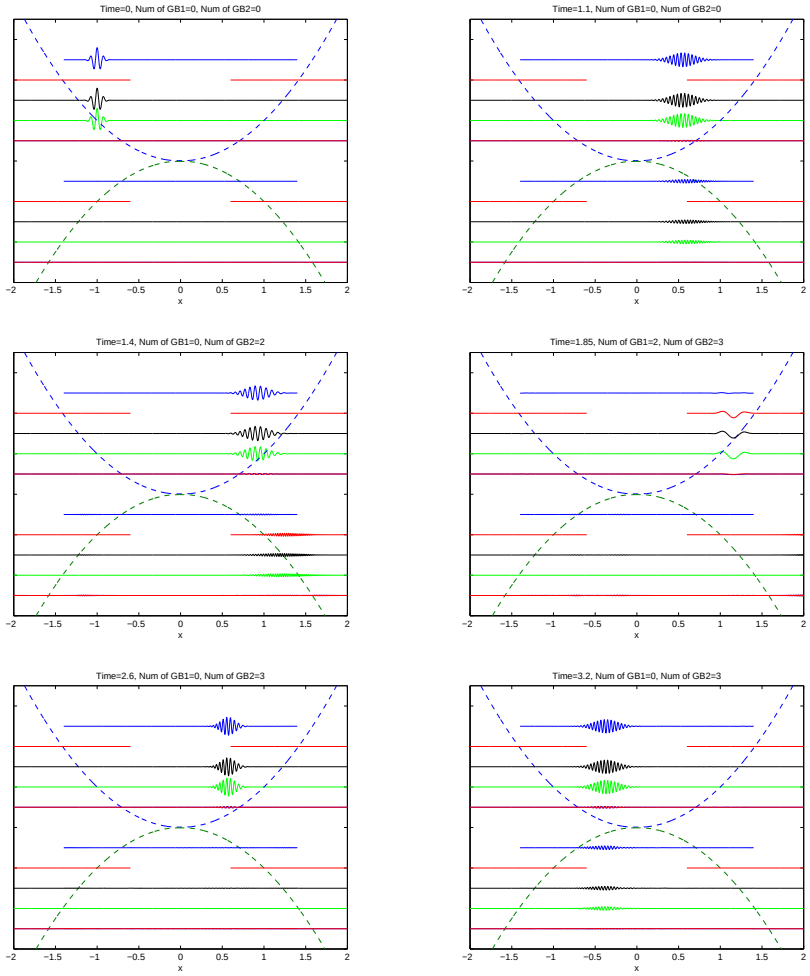

FIGURE 7. Example 3: The results using the hybrid method and TSSP. For each function, we only draw its real part. The imaginary part looks similar. In every picture, there are two solutions $\psi_{1,2}$ represented by two sets of curves with ψ_1 on the upper half and ψ_2 on the lower half. In each set of curves, the upper most one is by TSSP on DSch, i.e. $F_{1,2}(x,\,t)$. The next is $\tilde{G}_{1,2}(x,\,t)$. Then $F_{1,2}(x,\,t)+\tilde{G}_{1,2}(x,\,t)$. The fourth one is the solution by TSSP. We consider it as the reference solution, $\psi_{1,2}(x,\,t)$. The fifth is the difference between $\psi_{1,2}(x,\,t)$ and $\Big(F_{1,2}(x,\,t)+\tilde{G}_{1,2}(x,\,t)\Big)$. The dashed curves on the background demonstrate the the shape of the potential energies.

6. **Conclusion.** In this paper, we developed a hybrid method that couples a Schrödinger solver with the Gaussian beam method for quantum tunnelings through potential barriers or electronic potential energy surfaces. The idea is to use GB as much as possible and then solve the Schrödinger equation or the system of the

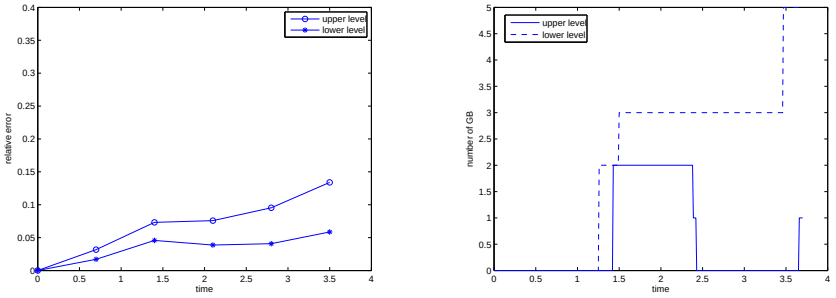

FIGURE 8. Example 3: The relative error, $\text{error}_1(t) + \text{error}_2(t)$ (left) and the numbers of Gaussian Beams (right) of the hybrid method.

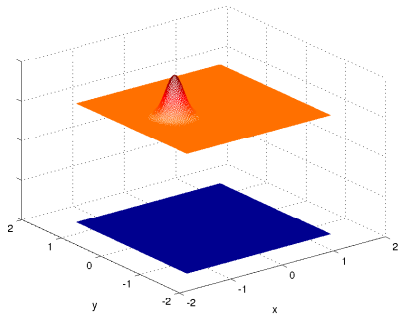

FIGURE 9. Example 4: The initial data at $t = 0$. The upper one is $|\psi_1(\mathbf{x}, t)|$ and the lower one $|\psi_2(\mathbf{x}, t)|$.

Schrödinger equations near potential barriers or the surface hopping areas. Buffer zones are used to convert the data between the Schrödinger and the Gaussian beam solutions. Several numerical examples in both one and two space dimensions are given to show that the hybrid method captures the quantum tunneling with quantum simulation only locally around the barriers or hopping zones, thus is much more efficient than a direct quantum solver in the entire domain.

REFERENCES

[1] M. Baer, "Beyond Born-Oppenheimer: Electronic Nonadiabatic Coupling Terms and Conical Intersections," Wiley, Hoboken, New Jersey, 2006.

[2] W. Bao, S. Jin and P. Markowich, *On the time-splitting spectral approximations for the Schrödinger equation in the semiclassical regime*, J. Comput. Phys., **175** (2002), 487–524.

[3] M. Born and R. Oppenheimer, *Zur Quantentheorie der Molekeln*, Ann. Phys., **84** (1927), 457–484.

[4] N. Ben Abdallah, P. Degond and I. M. Gamba, *Coupling one-dimensional time-dependent classical and quantum transport models*, J. Math. Phys., **43** (2002), 1–24.

[5] M. V. Berry, *Quantal phase factors accompanying adiabatic changes*, Proc. R. Soc. Lond Ser. A, **392** (1984), 45–57.

[6] V. Cerveny, M. M. Popov and I. Psencik, *Computation of wave fields in inhomogeneous media-Gaussian beam approach*, Geophys. J. R. Astr. Soc., **70** (1982), 109–128.

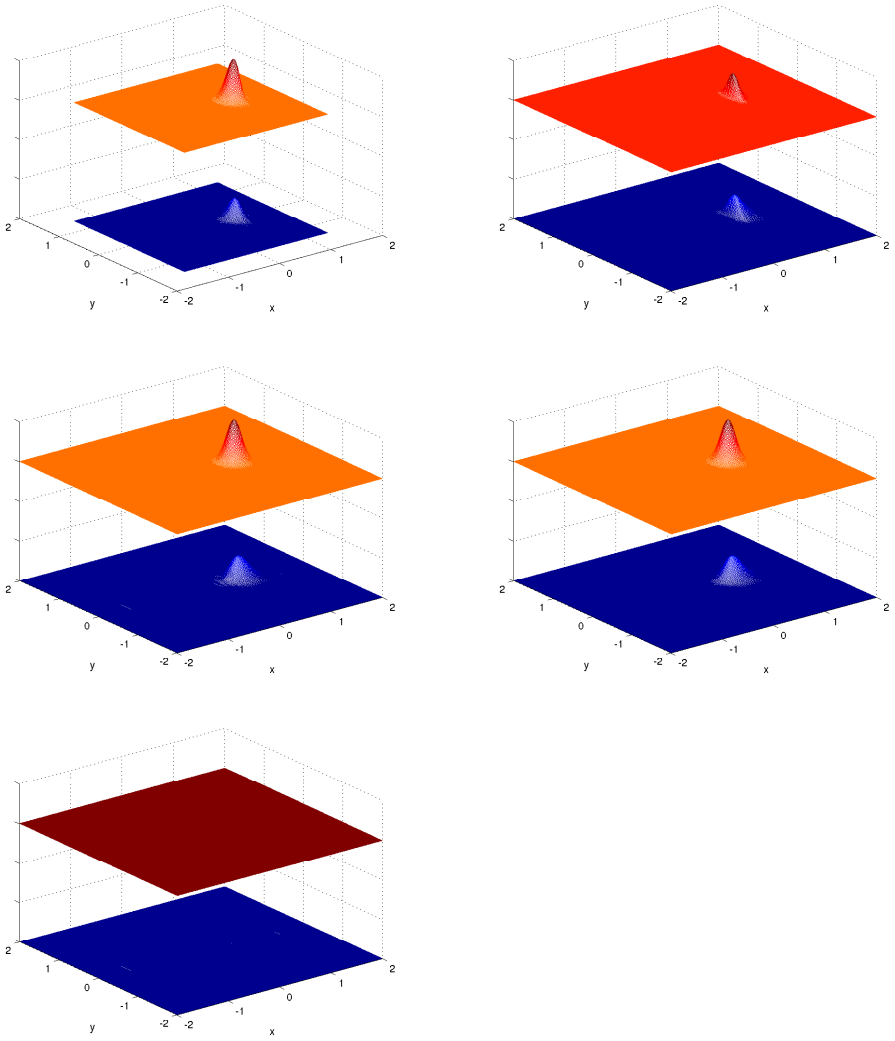

FIGURE 10. Example 4 (a): $t = 0.75$. The results using the hybrid method and TSSP. For each function, we only draw its modulus. The upper one in each picture is for $\psi_1(\mathbf{x}, t)$ and the lower one $\psi_2(\mathbf{x}, t)$. The picture on the left in the top row is the solutions on DSch using TSSP, i.e., $F_{1,2}(\mathbf{x}, t)$. The picture on the right in the top row is $\tilde{G}_{1,2}(\mathbf{x}, t)$. The picture on the left in the middle row is the summations of the previous two, i.e., $F_{1,2}(\mathbf{x}, t) + \tilde{G}_{1,2}(\mathbf{x}, t)$. The picture on the right in the middle row is the reference solutions $\psi_{1,2}(\mathbf{x}, t)$. The picture on the bottom is the errors between $\psi_{1,2}(\mathbf{x}, t)$ and $\Big(F_{1,2}(\mathbf{x}, t) + \tilde{G}_{1,2}(\mathbf{x}, t)\Big)$.

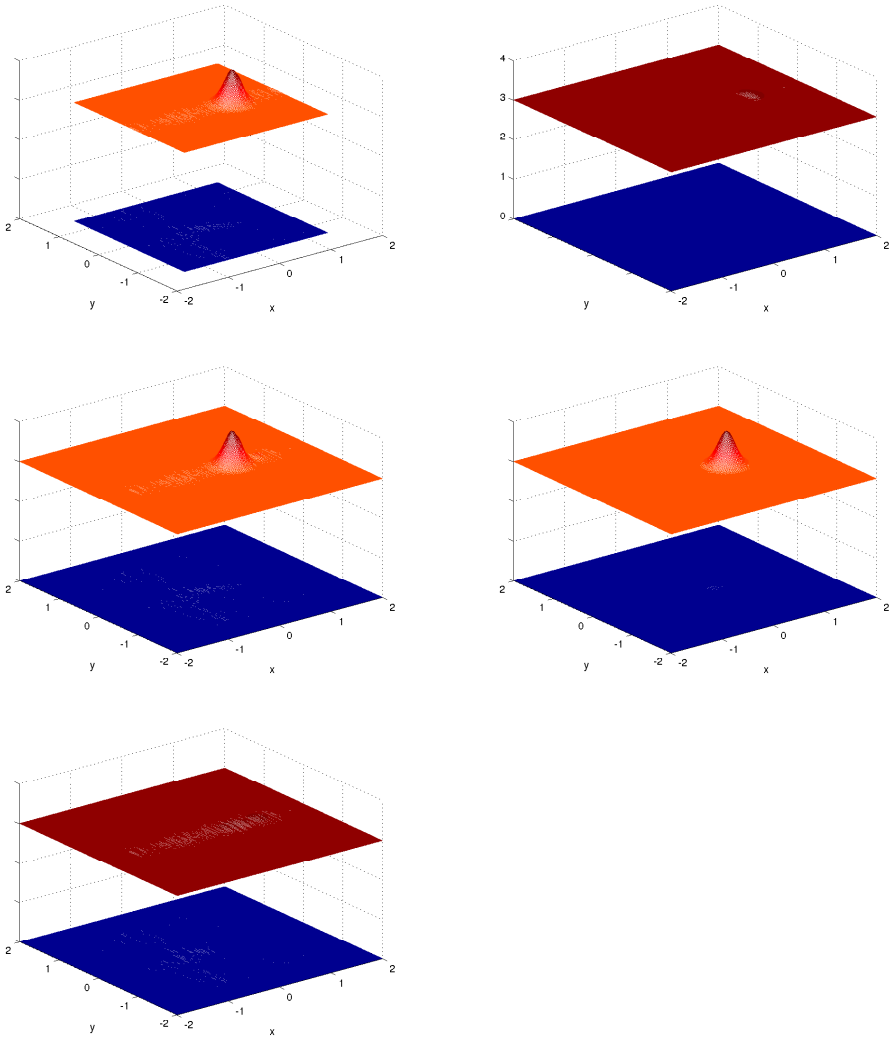

FIGURE 11. Example 4 (b): $t = 2.2$. The results using the hybrid method and TSSP.

[7] K. Drukker, *Basics of surface hopping in mixed quantum/classical simulations*, J. Comp. Phys., **153** (1999), 225–272.

[8] P. Gérard, P. A. Markowich, N. J. Mauser and F. Poupaud, *Homogenization limits and Wigner transforms*, Communications on Pure and Applied Mathematics, **50** (1997), 323–379.

[9] E. J. Heller, *Cellular dynamics: A new semiclassical approach to time-dependent quantum mechanics*, J. Chem. Phys., **94** (1991), 2723–2729.

[10] N. R. Hill, *Gaussian beam migration*, Geophys., **55** (1990), 1416–1428.

[11] I. Horenko, C. Salzmann, B. Schmidt and C. Schutte, *Quantum-classical Liouville approach to molecular dynamics: Surface hopping Gaussian phase-space packets*, J. Chem. Phys., **117** (2002), 11075–11088.

[12] S. Jin, P. Markowich and C. Sparber, *Mathematical and computational methods for semiclassical Schrödinger equations*, Acta Numerica, **20** (2011), 211–289.

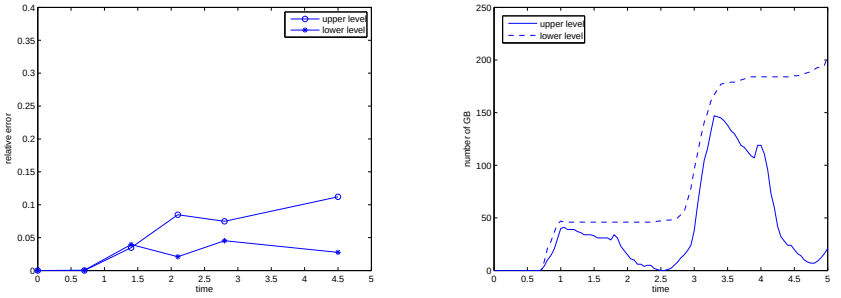

FIGURE 12. Example 4: The relative error, $\text{error}_1(t) + \text{error}_2(t)$ (left) and the numbers of Gaussian Beams (right) of the hybrid method.

[13] S. Jin and K. Novak, *A semiclassical transport model for thin quantum barriers*, Multiscale Modeling and Simulation, **5** (2006), 1063–1086.

[14] S. Jin and K. Novak, *A semiclassical transport model for two-dimensional thin quantum barriers*, J. Comp. Phys., **226** (2007), 1623–1644.

[15] S. Jin and K. Novak, *A coherent semiclassical transport model for pure-state quantum scattering*, Comm. Math. Sci., **8** (2010), 253–275.

[16] S. Jin, P. Qi and Z. Zhang, *An Eulerian surface hopping method for the Schrödinger equation with conical crossings*, SIAM Multiscale Modeling & Simulation, **9** (2011), 258–281.

[17] S. Jin, H. Wu and X. Yang, *Gaussian beam methods for the Schrödinger equation in the semi-classical regime: Lagrangian and Eulerian formulations*, Comm. Math. Sci., **6** (2008), 995–1020.

[18] S. Jin and D. Yin, *Computation of high frequency wave diffraction by a half plane via the Liouville equation and geometric theory of diffraction*, Communications in Computational Physics, **4** (2008), 1106–1128.

[19] L. Landau, *Zur Theorie der Energiebertragung II*, Physics of the Soviet Union, **2** (1932), 46–51.

[20] C. F. Kammerer and C. Lasser, *Wigner measures and codimension two crossings*, Jour. Math. Phys., **44** (2003), 507–527.

[21] C. F. Kammerer and C. Lasser, *Single switch surface hopping for molecular dynamics with transitions*, J. Chem. Phys., **128** (2008), 5–9.

[22] C. Lasser, T. Swart and S. Teufel, *Construction and validation of a rigorous surface hopping algorithm for conical crossings*, Comm. Math. Sci., **5** (2007), 789–814.

[23] S. Leung, J. Qian and R. Burridge, *Eulerian Gaussian beams for high frequency wave propagation*, Geophysics, **72** (2007), 61–76.

[24] J. Lu and X. Yang, *Frozen Gaussian approximation for high frequency wave propagation*, Commun. Math. Sci., **9** (2011), 663–683.

[25] M. Motamed and O. Runborg, *Taylor expansion and discretization errors in Gaussian beam superposition*, Wave Motion, **47** (2010), 421–439.

[26] W. H. Miller and Thomas F. George, *Semiclassical theory of electronic transitions in low energy atomic and molecular collisions involving several nuclear degrees of freedom*, Journal of Chemical Physics, **56** (1972), 5637–5652.

[27] B. N. Parlett, "The Symmetric Eigenvalue Problem," Classics ed., SIAM, Philadelphia, PA, USA, 1998.

[28] M. M. Popov, *A new method of computation of wave fields using Gaussian beams*, Wave Motion, **4** (1982), 85–97.

[29] J. Qian and L. Ying, *Fast Gaussian wavepacket transforms and Gaussian beams for the Schrödinger equation*, J. Comput. Phys., **229** (2010), 7848–7873.

[30] J. Ralston, *Gaussian beams and the propagation of singularities*, in "Studies in PDEs," MAA. Stud. Math., **23**, Math. Assoc. America, Washington, DC, (1982), 206–248.

[31] D. Sholla and J. Tully, *A generalized surface hopping method*, J. Chem. Phys., **109** (1998), 7702.
[32] J. Tully and R. Preston, *Trajectory surface hopping approach to nonadiabatic molecular collisions: The reaction of H^+ with D_2*, J. Chem. Phys., **55** (1971), 562–572.
[33] J. Tully, *Molecular dynamics with electronic transitions*, J. Chem. Phys., **93** (1990), 1061.
[34] N. Tanushev, *Superpositions and higher order Gaussian beams*, Commun. Math. Sci., **6** (2008), 449–475.
[35] N. Tanushev, J. Qian and J. Ralston, *Mountain waves and Gaussian beams*, SIAM J. Multiscale Model. Simul., **6** (2007), 688–709.
[36] N. M. Tanushev, R. Tsai and B. Engquist, *Coupled finite difference-Gaussian beam method for high frequency wave propagation*, UCLA CAM Reports, (2010), 10–40.
[37] N. M. Tanushev, B. Engquist and R. Tsai, *Gaussian beam decomposition of high frequency wave fields,* Journal of Computational Physics, **228** (2009), 8856–8871.
[38] A. Voronin, J. Marques and A. Varandas, *Trajectory surface hopping study of the Li + Li_2 ($X^1\Sigma_g^+$) Dissociation reaction*, J. Phys. Chem. A, **102** (1998), 6057–6062.
[39] D. Wei and X. Yang, *Gaussian beam method for high frequency wave propagation in heterogeneous media*, preprint 2011.
[40] W. R. E. Weiss and G. A. Hagedorn, *Reflection and transmission of high freuency pulses at an interface*, Transport Theory and Statistical Physics, **14** (1985), 539–565.
[41] D. Yin and C. Zheng, *Gaussian beam formulations and interface conditions for the one-dimensional linear Schrödinger equation*, Wave Motion, **48** (2011), 310–324.
[42] D. Yin and C. Zheng, *Composite Gaussian beam approximation method for multi-phased wave functions,* preprint, 2011.
[43] C. Zener, *Non-adiabatic crossing of energy levels*, Proceedings of the Royal Society of London, Series A, **137** 1932, 692–702.

Received May 2011; revised October 2011.

E-mail address: jin@math.wisc.edu
E-mail address: pqi@math.wisc.edu