# FAST HUYGENS SWEEPING METHODS FOR SCHRÖDINGER EQUATIONS IN THE SEMI-CLASSICAL REGIME

SHINGYU LEUNG[†], JIANLIANG QIAN[‡], AND SUSANA SERNA[§]

**Abstract.** We propose fast Huygens sweeping methods for Schrödinger equations in the semi-classical regime by incorporating short-time Wentzel-Kramers-Brillouin-Jeffreys (WKBJ) propagators into Huygens' principle. Even though the WKBJ solution is valid only for a short time period due to the occurrence of caustics, Huygens' principle allows us to construct the global-in-time semi-classical solution. To improve the computational efficiency, we develop analytic approximation formulas for the short-time WKBJ propagator by using the Taylor expansion in time. These analytic formulas allow us to develop two classes of fast Huygens sweeping methods, among which one is posed in the momentum space, and the other is posed in the position space, and both of these methods are of computational complexity $O(N \log N)$ for each time step, where $N$ is the total number of sampling points in the $d$-dimensional position space. To further speed up these methods, we also incorporate the soft-thresholding sparsification strategy into our new algorithms so that the computational cost can be further reduced. The methodology can also be extended to nonlinear Schrödinger equations. One, two, and three dimensional examples demonstrate the performance of the new algorithms.

**Key words.** Fast Huygens sweeping method, eikonal equation, WKBJ, convolution, fast Fourier transform, Schrödinger equation.

**AMS subject classifications.** 65N30, 65M60.

**1. Introduction.** Consider the Schrödinger equation for a particle with unity mass

$$(1.1) \qquad \left(i\hbar\frac{\partial}{\partial t} - H\right)U \equiv i\hbar\, U_t - V(x)U + \frac{\hbar^2}{2}\Delta U = 0, \ \ x \in \mathcal{R}^d, \ \ t > t_0,$$

$$(1.2) \qquad U(x,t_0) = U_{t_0}(x)$$

where $H = -\frac{\hbar^2}{2}\Delta + V(x)$, the potential $V$ is real and smooth, and $\hbar \equiv h/2\pi$ with $h$ a small (scaled) Planck's constant. When $\hbar$ is small, the wave function $U(x,t)$ for the Schrödinger equation is highly oscillatory, and it is very costly to apply direct methods such as finite-difference methods to compute these wave functions as such methods require very fine meshes to resolve oscillations generated by the equation. Therefore, alternative methods such as asymptotic methods are sought to resolve these highly oscillatory wave functions in the semi-classical regime, where semi-classical refers to an asymptotic theory in which one part of a system is described quantum-mechanically whereas the other, such as a particle trajectory, is treated classically. In this paper, we propose novel methods, called fast Huygens sweeping methods, for solving the Schrödinger equation in the semi-classical regime by incorporating the short-time Wentzel-Kramers-Brillouin-Jeffreys (WKBJ) propagator into Huygens' principle.

To develop these new methods, we utilize the following WKBJ ansatz for the quantum wave function,

$$(1.3) \qquad\qquad U(x,t) \approx A(x,t)\exp\left(\frac{i\,\tau(x,t)}{\hbar}\right),$$

---

[†]Department of Mathematics, the Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (masyleung@ust.hk).

[‡]Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA (qian@math.msu.edu).

[§]Department of Mathematics, UCLA, Los Angeles, CA 90095, USA (serna@math.ucla.edu).

where $A(x,t)$ is the amplitude function and $\tau(x,t)$ is the phase function. Applying this ansatz to the Schrödinger equation and considering the leading order singularities, we have the following eikonal equation for the phase function and the transport equation for the amplitude function,

$$(1.4) \qquad \tau_t + V(x) + \frac{1}{2}|\nabla\tau|^2 = 0 \,,$$

$$(1.5) \qquad A_t + \nabla\tau \cdot \nabla A + \frac{1}{2}\Delta\tau A = 0.$$

The transport equation for the amplitude is weakly coupled to the eikonal equation in the sense that one must first solve the eikonal equation to provide related coefficients for the transport equation. Because the eikonal equation is a non-linear first-order equation, in general there does not exist a global smooth (classical) solution for the equation. The concept of viscosity solution singles out a unique Lipschitz continuous solution for the eikonal equation among many possible generalized solutions. However, the gradient of the resulting viscosity solution for the eikonal equation can be discontinuous at so-called kinks so that the needed eikonal-related coefficients in the transport equation are not well-defined; consequently, the WKBJ ansatz (1.3) might not be valid globally in time in terms of globally smooth eikonal and amplitude.

On the other hand, we do observe that under suitable initial conditions the eikonal equation (1.4) always has a smooth solution for a short-time period, and this observation in turn implies that the WKBJ ansatz will be valid for a short-time period. Therefore, the question is: "Can we obtain globally valid semi-classical solutions for the Schrödinger equation by making use of the short-time valid WKBJ solutions?" The answer lies in Huygens' principle, which states that at each instant in time, each point in a wave field acts as a source for radiating new waves, which interfere in such a way as to create the wave field at a later time. Mathematically, such an interference is achieved by an integration with respect to those source locations, and the kernel of the integration is the Green function parameterized by these sources for the Schrödinger equation. Then a natural question is: "how to compute the Green function for the Schrödinger equation?" We propose to utilize the short-time valid WKBJ ansatz to construct short-time valid Green functions. Finally, by partitioning a large time period into several short-time periods, Huygens' principle allows us to sweep through the global-in-time period to obtain a globally valid semi-classical solution for the Schrödinger equation.

We develop novel approaches for computing semi-classical solutions by using this new idea from different perspectives. At first, we can construct short-time valid approximate asymptotic Green functions either numerically or analytically by using the Taylor expansion in time. Secondly, we can carry out interference integration either in the position space or in the momentum space. These different perspectives lead to at least four different implementations for constructing the semi-classical solution for the Schrödinger equation. Naturally, analytic Green functions result in much more efficient interference integration. Furthermore, interference integration carried out in the momentum space can be speed up by using FFTs directly while interference integration in the spatial domain can be speed up by FFT-based fast discrete convolutions. Eventually, analytic Green functions plus FFT based interference integrations lead to $O(N \log N)$ algorithms for each time step which eventually yield global-in-time semi-classical solutions to the Schrödinger equation, where $N$ is the total number of sampling points of the wave field in the $d$-dimensional position space. Therefore, we have named our new algorithms *fast Huygens sweeping methods* for the Schrödinger

equation in the semi-classical regime. To further speed up the proposed new algorithms, we also incorporate recently developed sparse dynamics into our algorithms. We also extend our methodology to a nonlinear Schrödinger equation.

**1.1. Related work.** Fast Huygens sweeping methods have been first proposed in [18] for Helmholtz equations in inhomogeneous media in the high-frequency regime, in which Huygens' principle is characterized by the Kirchhoff-Huygens integral, and the integration is carried out efficiently by constructing low-rank matrix approximations. We remark that the "sweeping" here has different meaning from that in the "fast sweeping methods" for solving static Hamilton-Jacobi equations in the literature.

Our methodology for developing fast Huygens sweeping method for the Schrödinger equation in the semi-classical regime is analogous to that for Helmholtz equations in the high frequency regime in that both methods try to take care of caustics automatically without identifying where caustics are, both appeal to Huygens' principle to obtain interference effects in wave fields, and both have very low-computational complexity. Therefore, we use the identical name for the two methods.

In terms of treating caustics, a popular approach is the Gaussian beam summation method [1, 26, 23, 13, 11, 9, 10, 31], and many efforts have been made to develop both efficient Lagrangian and Eulerian Gaussian beam methods[16, 20, 30, 14, 15, 17, 24, 25]. The first Eulerian Gaussian beam method has been proposed in [16], which was further developed in [14, 15, 17]. Efficient Lagrangian Gaussian beam methods based on fast wavepacket transforms have been proposed in [24] for Schrödinger equations and in [25] for wave equations. Convergence analysis of Gaussian beam summation methods has been carried out in [5, 2] in different contexts. Although they can treat caustics automatically, computationally Gaussian beam summation methods have some shortcomings related to exponential growth of beam width and expensive summation.

In terms of sparse dynamics for time-dependent PDEs, the general idea is to view the PDE solution in a transformed domain, where the information of the solution is encoded into coefficients corresponding to the transform basis. By choosing an appropriate basis the critical information of the solution can be concentrated into a few significant coefficients related to those crucial modes capturing the important behavior of the solution. If that is the case, then we may carry out a thresholding process to keep only those significant coefficients and throw away those insignificant ones so that the overall behavior of the solution is captured and the computational cost is reduced. In general, there are two approaches for carrying out the thresholding process: hard thresholding and soft thresholding. Hard thresholding has been used in Gaussian beam summations for Schrödinger equations [14, 15, 24] and wave equations in [25], resulting in accurate and efficient summation methods. Soft thresholding has been first used for time dependent PDEs in [27], and the resulting sparse dynamics leads to efficient and accurate representations of computed numerical solutions in the spectral domain.

Because the Schrödinger equation generates new scales during the evolution process, the methodology of [27] cannot be applied here. On the other hand, we may still replace the hard-thresholding process as used in [14, 15, 24, 25] with the soft-thresholding in the evolution process; consequently, although we are using the soft-thresholding strategy to sparsify the solution in the spectral (frequency/momentum) space, our methodology is different from the one in [27]. Nevertheless, such sparsification still results in significant speedup in the computation.

The rest of the paper is organized as follows. In Section 2, we summarize Huygens'

principle and incorporate the short-time WKBJ propagator into the superposition principle. In Section 3, we develop fast Huygens sweeping methods to implement Huygens' principle in both momentum and position spaces in the semi-classical regime for linear and nonlinear Schrödinger equations. In Section 4, we show numerical examples to demonstrate the efficiency and accuracy of the new algorithms.

**2. Semi-classical Green's function and Huygens' principle.** The Green function $G(x, t; x_0, t_0)$ satisfies the following partial differential equation [28]

$$(2.6) \qquad \left(i\hbar\frac{\partial}{\partial t} - H\right) G(x, t; x_0, t_0) = i\hbar\delta(x - x_0)\delta(t - t_0), \quad x \in \mathbb{R}^d, \quad t \geq t_0,$$

$$(2.7) \qquad G(x, t; x_0, t_0) = 0, \quad x \in \mathbb{R}^d, \quad t < t_0,$$

where $(x_0, t_0)$ are parameters, and the Hamiltonian operator $H$ takes the form of kinetic-plus-potential form: $H = -\frac{\hbar^2}{2}\frac{\partial^2}{\partial x^2} + V(x)$. By using Duhamel's principle, the above inhomogeneous formulation can be reduced to the following homogeneous initial value problem:

$$(2.8) \qquad \left(i\hbar\frac{\partial}{\partial t} - H\right) G(x, t; x_0, t_0) = 0, \quad x \in \mathbb{R}^d, \quad t > t_0,$$

$$(2.9) \qquad \lim_{t \to t_0^+} G(x, t; x_0, t_0) = \delta(x - x_0), \quad x \in \mathbb{R}^d,$$

$$(2.10) \qquad G(x, t; x_0, t_0) = 0, \quad x \in \mathbb{R}^d, \quad t < t_0.$$

Therefore, $G(x, t; x_0, t_0)$ can be seen as the response at position $x$ and time $t$ due to a point source at position $x_0$ and time $t_0$. According to Huygens' principle, the wave function $U(x, t)$ for $t > t_0$ for the Schrödinger equation can be written as

$$(2.11) \qquad U(x, t) = \int_{\mathbb{R}^d} G(x, t; x_0, t_0)U(x_0, t_0)dx_0, \quad t > t_0,$$

which formalizes the fact that the superposition of waves radiating from each point of an old wave creates a new wave at a later time, and the Green function provides appropriate weighting factors for the superposition. The above facts are well known in quantum mechanics; see [28].

To utilize Huygens' principle, we need to know the Green function, which requires solving the initial value problem (2.8) for the Schrödinger equation. Since we are interested in semi-classical solutions for the Schrödinger equation, we propose to compute the Green function asymptotically. It is well known that the $\delta$-function admits the following plane wave decomposition,

$$(2.12) \qquad \delta(x - x_0) = \left(\frac{1}{2\pi\hbar}\right)^d \int_{\mathbb{R}^d} e^{\frac{i(x - x_0)\cdot\xi}{\hbar}}d\xi;$$

namely, the initial condition for the Green's function is actually a superposition of plane waves. Hence to obtain needed ingredients in the asymptotic form of Green's functions, we will solve the following eikonal and transport equations:

$$(2.13) \qquad \tau_t + V(x) + \frac{1}{2}|\nabla\tau|^2 = 0, \quad t > t_0,$$

$$(2.14) \qquad \tau(x, t_0; \xi) = x \cdot \xi,$$

$$(2.15) \qquad A_t + \nabla\tau \cdot \nabla A + \frac{1}{2}\Delta\tau A = 0, \quad t > t_0,$$

$$(2.16) \qquad A(x, t_0; \xi) = 1,$$

where $\xi \in \mathcal{R}^d$ is a parameter, which can be viewed as a momentum variable as corresponding to $x$ as a position variable. According to the PDE theory on Hamilton-Jacobi equation, the eikonal equation (2.13) has a unique smooth solution for a short period of time; we denote this short period of time as $[t_0, t_0 + T]$, where $T > 0$ is a constant.

As a result, the eikonal $\tau(x, t; \xi)$ and the amplitude $A(x, t; \xi)$ exist and are smooth for $t_0 \leq t \leq t_0 + T$; accordingly,

$$\tilde{G}(x, t; \xi) \equiv A(x, t; \xi) e^{i \frac{\tau(x,t;\xi)}{\hbar}}$$

is a valid asymptotic solution for the Schrödinger equation with the plane wave initial condition,

$$\left(i\hbar \frac{\partial}{\partial t} - H\right) \tilde{G}(x, t; \xi) = 0, \quad x \in \mathbb{R}^d, \quad t \geq t_0, \tag{2.17}$$

$$\tilde{G}(x, t; \xi) = e^{ix \cdot \xi}, \quad x \in \mathbb{R}^d. \tag{2.18}$$

Next, to obtain the asymptotic Green function $G$ (without confusion still denoted as $G$), we assemble these computed ingredients into the following formula,

$$G(x, t; x_0, t_0) = \left(\frac{1}{2\pi\hbar}\right)^d \int_{\mathbb{R}^d} A(x, t; \xi) e^{\frac{i(\tau(x,t;\xi) - x_0 \cdot \xi)}{\hbar}} d\xi. \tag{2.19}$$

It is easy to check that the so-defined $G(x, t; x_0, t_0)$ satisfies the Schrödinger equation asymptotically in the time period $t_0 \leq t \leq t_0 + T$ and satisfies the corresponding point-source initial condition.

With the asymptotic Green function at our disposal, we can propagate an arbitrary initial wave function $U(x, t_0)$ for a short period of time,

$$U(x, t) = \int_{\mathbb{R}^d} G(x, t; x_0, t_0) U(x_0, t_0) dx_0, \quad t_0 < t \leq t_0 + T. \tag{2.20}$$

Now since the Hamiltonian is time-independent, the Green function satisfies the following property,

$$G(x, t; x_0, t_0) = G(x, t_1; x_0, t_2) \quad \text{if } t - t_0 = t_1 - t_2 > 0. \tag{2.21}$$

This implies that the short-time-valid Green function can be repeatedly used to propagate the wave function for long time,

$$U(x, t) = \int_{\mathbb{R}^d} G(x, t; x_0, t_k) U(x_0, t_n) dx_0, \quad t_k < t \leq t_k + T, \tag{2.22}$$

where $t_k = t_0 + k\,T$ for $k = 0, 1, 2, \cdots$. This way we may sweep through a long period of time so that we may obtain global-in-time asymptotic solutions for the Schrödinger equation.

The formulas (2.20) and (2.22) are formulated in the position space. We may also

obtain equivalent formulas posed in the momentum space,

$$
\begin{aligned}
U(x,t) &= \int_{\mathbb{R}^d} G(x,t;x_0,t_0)U(x_0,t_0)dx_0 \\
&= \left(\frac{1}{2\pi\hbar}\right)^d \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} A(x,t;\xi)e^{\frac{i(\tau(x,t;\xi)-x_0\cdot\xi)}{\hbar}}U(x_0,t_0)d\xi dx_0 \\
&= \left(\frac{1}{2\pi\hbar}\right)^d \int_{\mathbb{R}^d} A(x,t;\xi)e^{\frac{i\tau(x,t;\xi)}{\hbar}}\left[\int_{\mathbb{R}^d} U(x_0,t_0)e^{\frac{-ix_0\cdot\xi}{\hbar}}dx_0\right]d\xi
\end{aligned}
$$

$$
\text{(2.23)} \qquad = \int_{\mathbb{R}^d} \tilde{G}(x,t;\xi)\hat{U}(\xi,t_0)d\xi\,,
$$

where

$$
\text{(2.24)} \qquad \tilde{G}(x,t;\xi) = A(x,t;\xi)e^{\frac{i\tau(x,t;\xi)}{\hbar}}\,,
$$

$$
\text{(2.25)} \qquad \hat{U}(\xi,t_0) = \left(\frac{1}{2\pi\hbar}\right)^d \int_{\mathbb{R}^d} U(x_0,t_0)e^{\frac{-ix_0\cdot\xi}{\hbar}}dx_0\,.
$$

These equivalent formulations lead to different algorithms as we develop next. We notice that if the initial condition $U_0(x,t_0) \in S(\mathbb{R}^d)$ (the Schwartz class), the amplitude function $A(x,t;\xi) \in L^\infty$, and the phase function $\tau(x,t;\xi)$ is real, then the integrals defined in (2.20) and (2.23) converge absolutely and define functions in $L^\infty$; consequently, we may truncate the integral on $\mathbb{R}^d$ to be an integral on a bounded domain. Thus to avoid unnecessary technicality we will make these assumptions in the algorithmic development.

## 3. Fast Huygens sweeping methods: Algorithms.

**3.1. Straight-forward implementation in the position space.** We first consider the computational complexity of determining the solution $U(x,T)$ using (2.20), and a straight-forward implementation involves a phase-space integral which is expensive.

Let $n$ be the number of computational mesh on each physical $(x)$ and frequency $(\xi)$ dimension, $m$ be the number of secondary sources $(x')$ in each dimension, and $d$ be the dimension of the problem. Let $N = n^d$ and $M = m^d$. A naive implementation is as follows.

---

Algorithm 1 (Straight-forward implementation):
1. $\tau$ and $A$ are discretized in the physical space $(x)$ with $O(n^d)$ mesh points.
2. Since $\Delta t = O(\Delta x)$, the computational complexity for obtaining $\tau$ and $A$ for each initial condition (2.14) and (2.16) is $O(n^{d+1})$.
3. Obtain $\tau(x,t;\xi)$ and $A(x,t;\xi)$: There are $\xi = O(n^d)$ initial conditions. The total number of operations is $O(n^{2d+1})$.
4. Construct $G$ in (2.19): we need $O(n^d)$ operations for each $x$ ($O(n^d)$ in total) and $x'$ ($O(m^d)$ in total). The total number of operations is $O(n^{2d}m^d)$.
5. Construct $U$ in (2.20): we need $O(m^d)$ operations for each $x$ ($O(n^d)$ in total). The total number of operations is $O(n^d m^d)$.

---

Numerically, the eikonal equation and the amplitude equation can be solved efficiently using any well-developed high order numerical methods like WENO5-TVDRK3

[21, 12] or Power-ENO [29]. So, to conclude, the total number of operations for pre-processing (steps 1-4) is $O(n^{2d}m^d) = O(n^{3d})$ if $m = O(n)$. To compute $U(x, kT)$ for each $k = 1, 2, 3, \cdots$ we require extra $O(n^{2d})$ operations (step 5). Concerning the storage requirement, the algorithm requires declaring explicitly a matrix variable $G$ which requires $O(n^d m^d) = O(n^{2d})$ bytes. Once we have constructed the variable, we can store it in the order of $x'$ in the hard-drive. In the post-processing step when we construct $U$, we only need to declare a variable of $O(n^d)$ corresponding to $G$ associated to one $x_0$. For a typical two-dimensional computation, the preprocessing step requires $O(n^4)$ operations, while each post-processing step takes extra $O(n^2)$ operations. This simple calculation shows that the complexity of this straight-forward implementation is too high which makes it impractical for three-dimensional computation.

**3.2. Improved implementation in the momentum space.** In the straight-forward implementation we discussed above, one has to construct the matrix $G$ according to (2.19) which is the most expensive part of the method in terms of both computation and memory. Instead of performing this explicitly, we first note that the integral (2.20) can be re-written in the form (2.23). In other words, instead of formulating the problem by computing the propagator directly and then performing the integral in $x_0$, we can express the propagator in the frequency space and compute the integral in the $\xi$-variable.

Since the formula (2.23) only involves an integral in the momentum space, it provides us with an improved implementation. Because in the momentum space we have the fast Fourier transform (FFT), one advantage of such an approach is that it drops the computational complexity from $O(n^{2d})$ to $O(n^d \log n)$. We consider the computational complexity of determining the solution $U(x, T)$. To use FFT, we take $n = m$.

---

Algorithm 2 (Improved implementation in the momentum space):
1. $\tau$ and $A$ are discretized in the $x$ space with $O(n^d)$ mesh points.
2. Since $\Delta t = O(\Delta x)$, the computational complexity for obtaining $\tau$ and $A$ for each initial condition (2.14) and (2.16) is $O(n^{d+1})$.
3. Obtain $\tau(x, t; \xi)$ and $A(x, t; \xi)$: there are $O(n^d)$ initial conditions in terms of $\xi$. The total number of operations is $O(n^{2d+1})$.
4. Construct $\tilde{G}$ in (2.24): the total number of operations is $O(n^{2d})$.
5. Construct $\hat{U}$ in (2.25): the total number of operations is $O(n^d \log n)$ if FFT is used. Otherwise, $O(n^{2d})$ operations are needed.
6. Construct $U$ in (2.23): we need $O(n^d)$ operations for each $x$ ($O(n^d)$ in total). The total number of operations is $O(n^{2d})$.

---

So, to conclude, this simple rearrangement of integration can drop the total number of operations for preprocessing (steps 1-4) from $O(n^{3d})$ to $O(n^{2d+1})$. To compute $U(x, kT)$ for each $k = 1, 2, 3, \cdots$ we require extra $O(n^{2d})$ operations (steps 5-6).

Besides the computational efficiency, another advantage of such a formulation is in the memory requirement of the overall algorithm. Since the integral in (2.23) is done with respect to $\xi$, we can output $\tilde{G}$ to the memory right away for each individual $\xi$. This implies that we require only $O(n^d)$ bytes for the variables $A$ and $\tau$, even though $G$ itself takes $O(n^{2d})$ bytes.

**3.3. Analytic approximation to eikonals and amplitudes.** Apparently, the most expensive part corresponds to the computation of eikonals and amplitudes in carrying out Huygens' principle asymptotically. We propose to use analytic formulas to approximate eikonals and amplitudes. The idea is based on a short-time Taylor expansion, and similar ideas have been used for eikonals only in different contexts in [19, 8].

Assuming that $\Delta t$ is small, we can expand both amplitude and phase functions using the Taylor series in $\Delta t$,

$$A(x, \Delta t; \xi) = 1 + A_1(x, \xi)\Delta t + A_2(x, \xi)\Delta t^2 + O(\Delta t^3),$$

(3.26)     $$\tau(x, \Delta t; \xi) = x \cdot \xi + \tau_1(x, \xi)\Delta t + \tau_2(x, \xi)\Delta t^2 + O(\Delta t^3).$$

Now, we plug in these expressions into (1.4) to (1.5) and collect the same order terms to obtain

$$A_1(x, \xi) = 0, \quad A_2(x, \xi) = \frac{1}{4}\Delta V;$$

$$\tau_1(x, \xi) = -V(x) - \frac{1}{2}|\xi|^2, \quad \tau_2(x, \xi) = \frac{1}{2}\nabla V \cdot \xi.$$

Since these formulas are linked to the potential and the momentum directly, they provide us with a set of analytic approximation formulas for eikonals and amplitudes. With these at our disposal, we develop more efficient algorithms in Huygens' superposition in either the momentum or position space.

**3.4. Taylor-expansion based spectral method in the momentum space.** Taking only the zeroth order term of $A$ and the linear term of $\tau$, (2.23) becomes

$$
\begin{aligned}
U(x, \Delta t) &= \left(\frac{1}{2\pi\hbar}\right)^d \int_{\mathbb{R}^d} \tilde{G}(x, \xi, \Delta t)\hat{U}(\xi, 0)d\xi \\
&\simeq \left(\frac{1}{2\pi\hbar}\right)^d \int_{\mathbb{R}^d} \exp\left[\frac{i}{\hbar}\left\{x \cdot \xi - \left[V(x) + \frac{1}{2}|\xi|^2\right]\Delta t\right\}\right]\hat{U}(\xi, 0)d\xi \\
&= \left(\frac{1}{2\pi\hbar}\right)^d \exp\left[-\frac{i}{\hbar}V(x)\Delta t\right]\int_{\mathbb{R}^d} \exp\left[\frac{i}{\hbar}\left(x \cdot \xi - \frac{\Delta t}{2}|\xi|^2\right)\right]\hat{U}(\xi, 0)d\xi \\
&= \left(\frac{1}{2\pi\hbar}\right)^d \exp\left[-\frac{i}{\hbar}V(x)\Delta t\right]\int_{\mathbb{R}^d} \exp\left[\frac{i}{\hbar}(x \cdot \xi)\right]\exp\left[-\frac{i\Delta t}{2\hbar}|\xi|^2\right]\hat{U}(\xi, 0)d\xi
\end{aligned}
$$

(3.27)     $$\simeq \exp\left[-\frac{i}{\hbar}V(x)\Delta t\right]\mathcal{F}^{-1}\left[\exp\left(-\frac{i\Delta t}{2\hbar}|\xi|^2\right)\mathcal{F}(U)\right].$$

Similarly, the higher order version is as follows:

$$U(x, \Delta t) \simeq \left(1 + \frac{\Delta t^2}{4}\Delta V\right)\exp\left[-\frac{i}{\hbar}V(x)\Delta t\right]$$

(3.28)     $$\mathcal{F}^{-1}\left[\exp\left(-\frac{i\Delta t}{2\hbar}|\xi|^2\right)\mathcal{F}(U)\right]\left(x + \frac{\Delta t^2}{2}\nabla V\right).$$

Note that the lower order version of our scheme is essentially the same as the first order time-splitting spectral scheme discussed in [22, 3]. However, there are at least two differences between these two frameworks. The first is that our algorithm is derived based on the approximation of the WKBJ propagator, while the first order

time-splitting spectral scheme was derived by applying the operator splitting to the equation itself. Moreover, these methods are generalized to higher order by different mechanisms. The first order time-splitting spectral scheme in [22, 3] was generalized to high-order by Strang's splitting, while our approach extends to high-order by incorporating more terms in the Taylor's expansion.

## 3.5. Taylor-expansion based convolution methods in the position space.

### 3.5.1. Low-order and High-order methods. Next, we use (2.20) in the position space. We approximate the asymptotic Green function by

$$
G(x, \Delta t; x', 0) \simeq \left(\frac{1}{2\pi\hbar}\right)^d \int_{\mathbb{R}^d} e^{\frac{i}{\hbar}\left[x\cdot\xi - \left(V(x) + \frac{1}{2}|\xi|^2\right)\Delta t - x'\cdot\xi\right]} d\xi
$$

$$
(3.29) \qquad = \frac{1}{(i2\pi\hbar\Delta t)^{d/2}} \exp\left[\frac{-i}{\hbar}V(x)\Delta t\right] \exp\left[\frac{i}{2\hbar\Delta t}|x - x'|^2\right].
$$

As a result, the integral (2.20) can be approximated by

$$
U(x, \Delta t)
$$

$$
= \int_{\mathbb{R}^d} G(x, \Delta t; x', 0)U(x', 0)dx'
$$

$$
(3.30) \quad \simeq \frac{1}{(i2\pi\hbar\Delta t)^{d/2}} \exp\left[\frac{-i}{\hbar}V(x)\Delta t\right] \int_{\mathbb{R}^d} \exp\left[\frac{i}{2\hbar\Delta t}|x - x'|^2\right] U(x', 0)dx'.
$$

We can take advantage of the special structure to compute the convolution efficiently using FFT. For simplicity we only discuss the numerical procedure in 1D and it is straight-forward to extend the approach to higher dimensions.

We first approximate the integral on a uniform mesh $x_i$ using the Trapezoidal rule, i.e.

$$
(3.31)
$$
$$
U(x_i, \Delta t) = \frac{1}{(i2\pi\hbar\Delta t)^{1/2}} \exp\left[\frac{-i}{\hbar}V(x_i)\Delta t\right] \Delta x \sum_j \exp\left[\frac{i}{2\hbar\Delta t}|x_i - x_j|^2\right] U(x_j, 0).
$$

In the form of matrix-vector multiplication, we denote the summation by a symmetric Toeplitz matrix $\mathbf{W}$ with each entry given by $W_{i,j} = \exp\left[\frac{i}{2\hbar\Delta t}|x_i - x_j|^2\right]$. To efficiently compute the multiplication in (3.31), we extend the $n$-by-$n$ symmetric Toeplitz matrix into a $2n$-by-$2n$ cyclic Toeplitz matrix $\tilde{\mathbf{W}}$ with the first row given by

$$
\tilde{\mathbf{W}}_1 = [W_{1,1} \ W_{1,2} \ \cdots \ W_{1,n} \ 0 \ W_{1,n} \ W_{1,n-1} \ \cdots \ W_{1,3} \ W_{1,2}],
$$

and also extend the vector $\mathbf{U} = [U_1 \ U_2 \ \cdots \ U_n]^T$ to $\tilde{\mathbf{U}} = [U_1 \ U_2 \ \cdots \ U_n \ 0 \ \cdots \ 0]^T \in \mathbb{R}^{2n}$. Since the cyclic matrix can be diagonalized by the discrete Fourier matrix, the product $\mathbf{W}\mathbf{U}$ is the first $n$ elements of $\mathcal{F}^{-1}\left[\mathcal{F}(\tilde{\mathbf{W}}_1) \cdot \mathcal{F}(\mathbf{U})\right]$, where $\mathcal{F}$ is the Fourier transform operator and $\mathcal{F}^{-1}$ denotes the inverse Fourier transform.

The computational complexity for obtaining $U(x_i, \Delta t)$ in (3.31) is summarized here:

---

Algorithm 3 (Low-order Taylor-Expansion Based Convolution Method):

1. Set $k = 0$ and compute the Fourier coefficients $\mathcal{F}(\tilde{\mathbf{W}}_1)$: the total number of operations is $O(n^d \log n)$.
2. $k = k + 1$.
3. Compute $\mathcal{F}(\tilde{\mathbf{U}})$ and determine the product $\mathcal{F}(\tilde{\mathbf{W}}_1) \cdot \mathcal{F}(\tilde{\mathbf{U}})$: the total number of operations is $O(n^d \log n)$.
4. Apply the inverse FFT $\mathcal{F}^{-1}\left[\mathcal{F}(\tilde{\mathbf{W}}_1) \cdot \mathcal{F}(\tilde{\mathbf{U}})\right]$ and keep only the first $n$ elements: the total number of operations is $O(n^d \log n)$.
5. Multiply the result by $(i2\pi\hbar\Delta t)^{-d/2} \exp\left[\frac{-i}{\hbar}V(x_i)\Delta t\right](\Delta x)^d$: the total number of operations is $O(n^d)$.
6. Go to step 2 until $t_k = t_f$.

We can regard step 1 as an preprocessing step which takes only $O(n^d \log n)$ operations. Considering the propagation step to obtain each $U(x, k\Delta t)$ for $k = 1, 2, \cdots$, we find that this method drops the computational complexity from $O(n^{2d})$ in Algorithm 2 to only $O(n^d \log n)$.

Numerically, $\Delta t$ in this approximation cannot be arbitrarily chosen. To resolve the oscillations in the coefficients of $W_{i,j}$, we require that the phase difference between $W_{1,n-1}$ and $W_{1,n}$ should be less than $2\pi$, i.e. we require

$$\frac{[(n-1)^2 - (n-2)^2]\Delta x^2}{2\hbar\Delta t} = \alpha 2\pi$$

for some $0 < \alpha < 1$, which implies that

$$(3.32) \qquad \Delta t = \frac{(2n-3)\Delta x^2}{\alpha 4\pi\hbar} = O\left(\frac{\Delta x}{\hbar}\right).$$

Note that this constraint imposes a **lower** bound on the marching step size $\Delta t$. For a given $\hbar$ and $\Delta x$, the method requires one to pick a large enough $\Delta t$ in order to resolve the oscillations in $W_{i,j}$. Indeed the larger the value of $\Delta t$, the faster we reach the final solution. However, we have to control the error introduced in the Taylor approximation at the same time. Therefore, we pick $\alpha$ close to, but smaller than, 1.

High-order generalization of our approach is possible as well. We can simply add in more terms in the Taylor approximation. For example, we consider again the one-dimensional case in detail. Taking the second order Taylor expansion for both amplitude and phase functions, we have the Green function,

$$G(x, \Delta t; x') \simeq \left(\frac{1}{i2\pi\hbar\Delta t}\right)^{1/2}\left(1 + \frac{\Delta t^2}{4}V''\right)\exp\left[\frac{-i}{\hbar}\Delta t\, V\right]$$

$$(3.33) \qquad \exp\left[\frac{i}{2\hbar\Delta t}\left|x - x' + \frac{\Delta t^2}{2}V'\right|^2\right]$$

$$= \left(\frac{1}{i2\pi\hbar\Delta t}\right)^{1/2}\left(1 + \frac{\Delta t^2}{4}V''\right)\exp\left[\frac{-i}{\hbar}\left(\Delta t\, V - \frac{\Delta t^3}{8}|V'|^2\right)\right]$$

$$(3.34) \qquad \exp\left[\frac{i}{2\hbar\Delta t}|x - x'|^2\right]\exp\left[\frac{i\Delta t}{2\hbar}(x - x')\cdot V'\right].$$

Note, however, that the extra term $(x - x')\cdot V'$ destroys the Toeplitz property of the $n$-by-$n$ matrix $\mathbf{W}$ and we cannot directly apply the FFT to construct $U$ according to (2.20).

To develop efficient numerical algorithm for the convolution, we first introduce a new variable

$$y = x + \frac{\Delta t^2}{2} V'(x)$$

so that the solution at $\Delta t$ is now given by

$$U(x, \Delta t) \simeq \frac{1}{\sqrt{i2\pi\hbar\Delta t}} \left(1 + \frac{\Delta t^2}{4} V''\right) \exp\left[\frac{-i}{\hbar} V \Delta t\right] \int_{x'} \exp\left[\frac{i}{2\hbar\Delta t}|y - x'|^2\right] U(x', 0) dx'.$$

Numerically, we first obtain the integral

$$I(y) = \int_{x'} \exp\left[\frac{i}{2\hbar\Delta t}|y - x'|^2\right] U(x', 0) dx'$$

on a uniform mesh $y = y_i$ using the same FFT algorithm as in the Algorithm 3. Next we interpolate the solution onto the non-uniform mesh $X_i^* = x_i + \frac{\Delta t^2}{2} V'(x_i)$ and then finally we assign the solution

(3.35) $$U(x_i, \Delta t) = \frac{1}{\sqrt{i2\pi\hbar\Delta t}} \left(1 + \frac{\Delta t^2}{4} V''(x_i)\right) \exp\left[\frac{-i}{\hbar} V(x_i)\Delta t\right] I(X_i^*).$$

To summarize, we have the following algorithm based on the high-order Taylor expansion:

---

Algorithm 4 (High-order Taylor Expansion Based Convolution Method):
1. Set $k = 0$ and compute the Fourier coefficients $\mathcal{F}(\tilde{\mathbf{W}}_1)$: the total number of operations is $O(n^d \log n)$.
2. $k = k + 1$.
3. Compute $\mathcal{F}(\tilde{\mathbf{U}})$ and determine the product $\mathcal{F}(\tilde{\mathbf{W}}_1) \cdot \mathcal{F}(\tilde{\mathbf{U}})$: the total number of operations is $O(n^d \log n)$.
4. Apply the inverse FFT $\mathcal{F}^{-1}\left[\mathcal{F}(\tilde{\mathbf{W}}_1) \cdot \mathcal{F}(\tilde{\mathbf{U}})\right]$ and keep the first $n$ elements: the total number of operations is $O(n^d \log n)$.
5. Interpolate the solution on the mesh $X_i^* = x_i + \frac{\Delta t^2}{2} \nabla V(x_i)$: the total number of operations is $O(n)$.
6. Multiply the result by $(i2\pi\hbar\Delta t)^{-d/2} \left(1 + \frac{\Delta t^2}{4} \Delta V(x_i)\right) \exp\left[\frac{-i}{\hbar} V(x_i)\Delta t\right] (\Delta x)^d$: the total number of operations is $O(n^d)$.
7. Go to step 2 until $t_k = t_f$.

---

**3.5.2. Energy conservation.** One important physical quantity is the position density of the wave function which can be interpreted as the probability density function of locating a particle. In particular, we have

$$\int_{x \in \mathbb{R}^d} |U(x, t)|^2 dx = 1$$

for all $t \geq 0$. In this section, we consider the above low-order scheme (3.30) and the high-order scheme (3.35) and discuss the convergence to this quantity.

THEOREM 3.1. *The low-order Taylor-approximation based scheme Algorithm 3 (3.30) and the high-order Taylor-approximation based scheme Algorithm 4 (3.35) satisfy*

$$\int_x |U(x, \Delta t)|^2 dx = \int_x |U(x, 0)|^2 dx \quad and$$

$$\int_x |U(x, \Delta t)|^2 dx = \left[1 + O\left(\Delta t^4\right)\right] \int_x |U(x, 0)|^2 dx\,,$$

*respectively.*

*Proof.* We consider one-dimensional cases. Let $\alpha = (i2\pi\hbar\Delta t)^{-1/2}$ so that $|\alpha|^2 = \alpha\bar{\alpha} = (2\pi\hbar\Delta t)^{-1}$.

$$\int_x |U(x, \Delta t)|^2 dx = \int_x U(x, \Delta t)\overline{U(x, \Delta t)}dx$$

$$= \int_x |\alpha|^2 \left\{\exp\left[\frac{-i}{\hbar}V(x)\Delta t\right] \int_{x'} \exp\left[\frac{i}{2\hbar\Delta t}|x - x'|^2\right] U(x', 0)dx'\right\}$$

$$\left\{\exp\left[\frac{i}{\hbar}V(x)\Delta t\right] \int_{y'} \exp\left[-\frac{i}{2\hbar\Delta t}|x - y'|^2\right] U(y', 0)dy'\right\} dx$$

$$= |\alpha|^2 \iint_{x' \times y'} \exp\left[\frac{i}{2\hbar\Delta t}(|x'|^2 - |y'|^2)\right]$$

$$\left\{\int_x \exp\left[-\frac{i}{\hbar\Delta t}|x' - y'| \cdot x\right] dx\right\} U(x', 0)\overline{U(y', 0)}dx'dy'$$

$$= \iint_{x' \times y'} \exp\left[\frac{i}{2\hbar\Delta t}(|x'|^2 - |y'|^2)\right] \delta(x' - y')U(x', 0)\overline{U(y', 0)}dx'dy'$$

$$= \int_x |U(x, 0)|^2 dx\,.$$

For the high-order scheme (3.35), we have $X^*(x) = x + \frac{\Delta t^2}{2}V'$ and

$$\int_x |U(x, \Delta t)|^2 dx = \int_x U(x, \Delta t)\overline{U(x, \Delta t)}dx$$

$$= \int_x |\alpha|^2 \left(1 + \frac{\Delta t^2}{4}V''\right)^2 \left\{\int_{x'} \exp\left[\frac{i}{2\hbar\Delta t}|X^* - x'|^2\right] U(x', 0)dx'\right\}$$

$$\left\{\int_{y'} \exp\left[-\frac{i}{2\hbar\Delta t}|X^* - y'|^2\right] \overline{U(y', 0)}dy'\right\} dx$$

$$= |\alpha|^2 \iint_{x' \times y'} \exp\left[\frac{i}{2\hbar\Delta t}(|x'|^2 - |y'|^2)\right]$$

$$\left\{\int_x \left(1 + \frac{\Delta t^2}{4}V''\right)^2 \exp\left[-\frac{i}{\hbar\Delta t}|x' - y'| \cdot X^*\right] dx\right\}$$

$$U(x', 0)\overline{U(y', 0)}dx'dy'\,.$$

Now, since

$$|\alpha|^2 \int_x \left(1 + \frac{\Delta t^2}{4} V''\right)^2 \exp\left[-\frac{i}{\hbar \Delta t}|x' - y'| \cdot X^*\right] dx$$

$$= |\alpha|^2 \int_x \left(1 + \frac{\Delta t^2}{2} V''\right) \exp\left[-\frac{i}{\hbar \Delta t}|x' - y'| \cdot X^*\right] dx + O\left(\Delta t^4\right)$$

$$= |\alpha|^2 \int_{X^*} \exp\left[-\frac{i}{\hbar \Delta t}|x' - y'| \cdot X^*\right] dX^* + O\left(\Delta t^4\right)$$

$$= \delta(x' - y') + O\left(\Delta t^4\right) ,$$

we obtain

$$\int_x |U(x, \Delta t)|^2 dx = \int_x U(x, \Delta t)\overline{U(x, \Delta t)} dx$$

$$= \iint_{x' \times y'} \exp\left[\frac{i}{2\hbar \Delta t}(|x'|^2 - |y'|^2)\right] \left[\delta(x' - y') + O\left(\Delta t^4\right)\right]$$

$$U(x', 0)\overline{U(y', 0)} dx' dy'$$

$$= \left[1 + O\left(\Delta t^4\right)\right] \int_x |U(x, 0)|^2 dx .$$

Now, at $t = t_f = k\Delta t$, we have

$$\int_x |U(x, t_f)|^2 dx = \left[1 + O\left(\Delta t^4\right)\right]^{t_f/\Delta t} \int_x |U(x, 0)|^2 dx .$$

Since $\Delta t = O(\Delta x)$, as $\Delta x \to 0$, we have

$$\lim_{\Delta x \to 0} \left[1 + O\left(\Delta t^4\right)\right]^{t_f/\Delta t} = 1$$

and so $\int_x |U(x, t_f)|^2 dx \to \int_x |U(x, 0)|^2 dx = 1$. □

**3.6. Sparse approximations.** Motivated by recent works in [14, 15, 24, 25, 27], we develop sparse evolution approaches to further speed up the above algorithms. To develop a sparse evolution approach, one needs to view the solution in the transformed domain, where the original solution has a sparse representation in the sense that only a few significant coefficients suffice to capture the overall behavior of the solution.

In [14, 15, 24, 25] the high frequency wave solutions are first represented in phase space by using a phase space transform, such as the wavepacket transform, then a hard-thresholding process is applied to the phase space representation so that a few modes in the phase space suffice to capture the solution; this strategy results in efficient high-frequency wave propagation.

In [27], the authors have proposed a new framework to efficiently approximate solutions to PDEs by sparse dynamics. The idea is to soft-threshold the solution in a certain basis in which the solution is *sparse*. Therefore, the complexity of the whole algorithm will depend on the number of non-zero terms retained in the sparse approximation which would be almost independent of the number of mesh points in representing the solution in the physical space. In all numerical examples demonstrated in that paper, the authors have considered solving various PDEs using the Fourier transform; i.e. one first applies the Fourier transform directly to the *equation*,

then soft-thresholds those updated spectral coefficients and updates the Fourier co-efficients according to the PDE in the spectral space, and finally inverse-transforms them back to the physical space.

Because the Schrödinger equation generates new scales during the evolution process, the methodology of [27] cannot be applied here. On the other hand, we may still replace the hard-thresholding process as used in [14, 15, 24, 25] with the soft-thresholding in the evolution process; consequently, although we are using the soft-thresholding operator, our methodology is different from the one in [27].

In the following, we adopt the soft-thresholding process to both the Taylor-expansion based spectral methods and the Taylor-expansion based convolution methods. Indeed, hard-thresholding like [14, 15, 24, 25] might also be used to improve the computational complexity but we will explore the idea in a future work.

To start with, we apply the soft-thresholding to the low-order expansion based Taylor spectral method, and this yields

$$(3.36) \qquad U(x, \Delta t) \simeq \exp\left[-\frac{i}{\hbar}V(x)\Delta t\right] \mathcal{F}^{-1}\left[\exp\left(-\frac{i\Delta t}{2\hbar}|\xi|^2\right)\tilde{\mathcal{F}}_\lambda(U)\right],$$

where $\mathcal{F}(U) = [\hat{u}_1\,\hat{u}_2\cdots\hat{u}_n]^T$ is the Fourier transform of $U$,

$$\tilde{\mathcal{F}}_\lambda(U) = [\mathcal{S}_\lambda(\hat{u}_1)\,\mathcal{S}_\lambda(\hat{u}_2)\cdots\mathcal{S}_\lambda(\hat{u}_n)]^T \text{ and } \mathcal{S}_\lambda[\hat{u}_i] = \max\left(|\hat{u}_i| - \lambda, 0\right)\frac{\hat{u}_i}{|\hat{u}_i|}.$$

Here $\mathcal{S}_\lambda$ is the so-called soft-thresholding operator which is applied to each of the Fourier coefficients, and the exponential factor is applied only to those non-zero $\tilde{U}(\xi_j, 0)$.

For high-order generalization, the procedure is similar to what we proposed in the previous section. One only needs to correct the magnitude by $\left(1 + \frac{\Delta t^2}{4}\Delta V\right)$ and interpolate the Fourier coefficients at $X^* = x + \frac{\Delta t^2}{2}\nabla V$, i.e.

$$U(x, \Delta t) \simeq \left(1 + \frac{\Delta t^2}{4}\Delta V\right)\exp\left[-\frac{i}{\hbar}V(x)\Delta t\right]$$

$$(3.37) \qquad \mathcal{F}^{-1}\left[\exp\left(-\frac{i\Delta t}{2\hbar}|\xi|^2\right)\tilde{\mathcal{F}}_\lambda(U)\right]\left(x + \frac{\Delta t^2}{2}\nabla V\right).$$

Here, we summarize Taylor expansion based algorithms in terms of sparse approximations.

---

Algorithm 5 (High-order Taylor-Expansion Based Spectral Methods With Sparse Approximation):

1. Set $k = 0$.
2. $k = k + 1$.
3. Compute $\mathcal{S}_\lambda[\mathcal{F}(U)]$: the total number of operations is $O(n^d \log n + n^d) = O(n^d \log n)$.
4. Multiply by $\exp\left(-\frac{i\Delta t}{2\hbar}|\xi|^2\right)$: the total number of operations is $O(J)$ where $J$ is the number of non-zero elements in $\mathcal{S}_\lambda[\mathcal{F}(U)]$.
5. Construct the solution at $t_k$ by applying inverse FFT and interpolating at $X^* = x + \frac{\Delta t^2}{2}\nabla V$: the number of operations is $\min[O(n^d J), O(n^d \log n)]$.
6. Multiply the result by $\left(1 + \frac{\Delta t^2}{4}\Delta V\right)\exp\left[\frac{-i}{\hbar}V(x_i)\Delta t\right]$: the total number of operations is $O(n^d)$.

7. Go to step 2 until $t_k = t_f$.

---

For some applications where the Fourier representation of the solution is sparse, i.e. $J \ll n$, we found that one could obtain an approximated solution in a similar precision as from Algorithm 3 using a $J$ as small as 0.2% of $n$. In practice on the other hand, high frequency modes can be generated in the Schrödinger equation so that the number of Fourier coefficients in the representation could grow in time. This implies that the constant $J$ or the parameter $\lambda$ has to be determined in an experiment-by-experiment fashion.

Application to the Taylor expansion convolution method is similar. We can simply apply the soft-thresholding to those Fourier coefficients. Note however that we are not applying the Fourier transform to the *partial differential equation* as in [27], but instead, these Fourier coefficients come from a convolution operator.

---

Algorithm 4' (High-order Taylor-Expansion Based Convolution Methods with Sparse Approximation):

1. Set $k = 0$ and compute the Fourier coefficients $\mathcal{F}(\tilde{\mathbf{W}}_1)$: the total number of operations is $O(n^d \log n)$.
2. $k = k + 1$.
3. Compute $\mathcal{S}_\lambda[\mathcal{F}(\tilde{\mathbf{U}})]$ and determine the product $\mathcal{F}(\tilde{\mathbf{W}}_1) \cdot \mathcal{S}_\lambda[\mathcal{F}(\tilde{\mathbf{U}})]$: the total number of operations is $O(n^d \log n + J)$, where $J$ is the number of non-zero elements in $\mathcal{S}_\lambda[\mathcal{F}(\tilde{\mathbf{U}})]$.
4. Compute the inverse FFT $\mathcal{F}^{-1}$ and keep only the first $n^d$ elements: the total number of operations is $\min[O(n^d J), O(n^d \log n)]$.
5. Interpolate the solution on the mesh $X_i^* = x_i + \frac{\Delta t^2}{2} \nabla V(x)$: the total number of operations is $O(n^d)$.
6. Multiply the result by $(i 2\pi \hbar \Delta t)^{-d/2} \left( 1 + \frac{\Delta t^2}{4} \Delta V(x_i) \right) \exp\left[ \frac{-i}{\hbar} V(x_i) \Delta t \right] (\Delta x)^d$: the total number of operations is $O(n^d)$.
7. Go to step 2 until $t_k = t_f$.

---

**3.7. Extension to nonlinear Schrödinger equations.** In this section, we propose a generalization of our above approach to approximate the solution to the nonlinear Schrödinger equation

$$i\hbar\, U_t - V(x)U + \frac{\hbar^2}{2}\Delta U + |U|^2 U = 0, \quad x \in \mathcal{R}^d, \ t > 0,$$

$$U(x,0) = A_0(x) \exp\left( \frac{i\,\tau_0(x)}{\hbar} \right).$$

To obtain the solution at $t = t_k$, the idea is to treat the nonlinear term semi-implicitly by freezing $|U|^2$ at $t = t_{k-1}$. This implies that we modify the potential by a time-dependent potential $V(x) - |U(x, t_{k-1})|^2$ and directly apply the above algorithm. For example, the low-order Taylor expansion scheme can be easily applied to the nonlinear Schrödinger equation and it is summarized as follows.

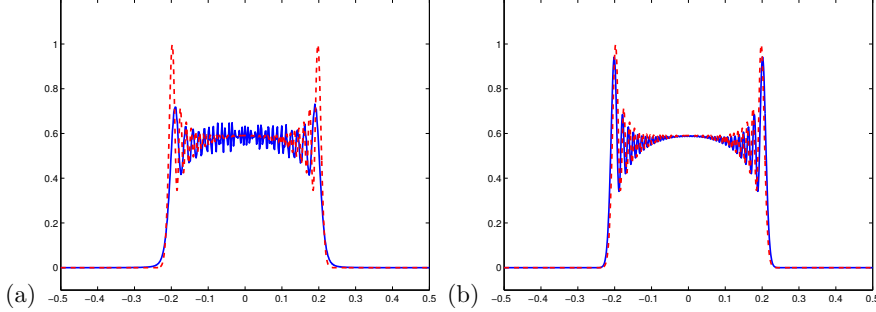---

Algorithm 6 (Nonlinear Schrödinger Equation):

FIG. 4.1. *(Section 4.1.1 Caustic with $\hbar = 1/1024$) Position density. Red dash line: Strang splitting spectral solution on a fine mesh. Blue solid line: Solution by the proposed method **Algorithm** 2. $\tau$ and $A$ are solved on a mesh with (a) $n_x = n_\xi = 513$ and (b) $n_x = n_\xi = 1025$.*

1. Set $k = 0$ and compute the Fourier coefficients $\mathcal{F}(\tilde{\mathbf{W}}_1)$: the total number of operations is $O(n^d \log n)$.
2. $k = k + 1$.
3. Compute $\mathcal{F}(\tilde{\mathbf{U}})$ and determine the product $\mathcal{F}(\tilde{\mathbf{W}}_1) \cdot \mathcal{F}(\tilde{\mathbf{U}})$: the total number of operations is $O(n^d \log n)$.
4. Apply the inverse FFT $\mathcal{F}^{-1}\left[\mathcal{F}(\tilde{\mathbf{W}}_1) \cdot \mathcal{F}(\tilde{\mathbf{U}})\right]$ and keep only the first $n$ elements: the total number of operations is $O(n^d \log n)$.
5. Multiply the result by $(i2\pi\hbar\Delta t)^{-d/2} \exp\left[\frac{-i}{\hbar}\{V(x_i) - |U(x_i, t_k)|^2\}\Delta t\right](\Delta x)^d$: the total number of operations is $O(n^d)$.
6. Go to step 2 until $t_k = t_f$.

**4. Numerical examples.** In this section, we will discuss the performance of various algorithms we proposed in section 3.2 to section 3.5 and in section 3.6. In all experiments below, for simplicity, we perform the numerical computations in a domain large enough such that the solution is of compact support. Therefore when computing the integrals, we need not worry about the influence from the wave function from outside the computational domain.

**4.1. Algorithm 2 from Section 3.2.**

**4.1.1. Caustics.** We first consider a simple one-dimensional example which leads to caustics using the usual WKBJ asymptotic ansatz. The background potential is given by $V(x) = 10$ and the initial wave function has the form

$$(4.38) \qquad U_0(x) = \exp\left(-25x^2\right)\exp\left[\frac{i\tau_0(x)}{\hbar}\right],$$

and the initial phase function $\tau_0(x)$ is given by

$$(4.39) \qquad \tau_0(x) = \frac{-1}{5}\ln\left[\exp(5x) + \exp(-5x)\right].$$

Figure 4.1 shows our computed solutions using **Algorithm 2** comparing to the solutions obtained by the Strang splitting spectral solution (red dash line) using a fine spatial mesh and a very small time stepping.
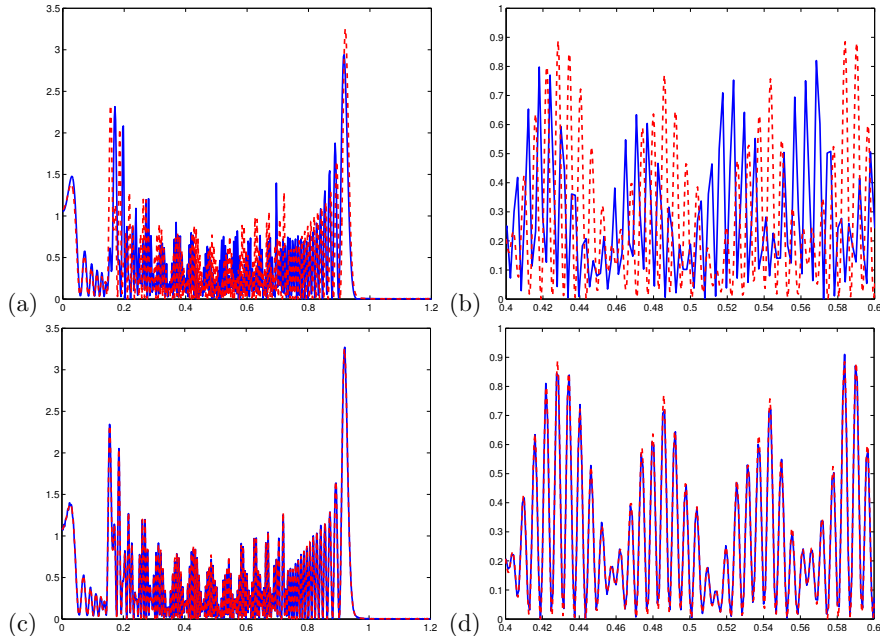
FIG. 4.2. *(Section 4.1.2 Cosine potential with $\hbar = 1/256\pi$) Position density. Red dash line: Spectral solution on a fine mesh. Blue solid line: Solution by the proposed method **Algorithm 2**. (a) $\tau$ and $A$ are solved on a mesh with $n_x = n_\xi = 2049$. (b) Zoom-in of (a). (c) $\tau$ and $A$ are solved on a mesh with $n_x = n_\xi = 4097$. (d) Zoom-in of (c).*

**4.1.2. A Gaussian in a cosine potential field.** We consider the following challenging example taken from [15] where the potential function is given by the cosine function

$$(4.40) \qquad V(x) = \frac{1}{8}\cos(2\pi x).$$

The initial profile is a Gaussian of standard deviation $\sigma_0 = 0.1$ centered at $x = 0$ with the zero initial momentum $p_0 = 0$:

$$(4.41) \qquad U_0(x) = \frac{1}{\sqrt{\sigma_0\sqrt{2\pi}}}\exp\left(-\frac{x^2}{4\sigma_0^2}\right)\exp\left(-\frac{ixp_0}{\hbar}\right).$$

Since the potential is a hill at $x = 0$, the Gaussian will be separated into two parts and each of them will fall into one of the potential wells centered at $x = \pm 0.5$, respectively.

In our previous work [15], we have demonstrated that it is important to reinitialize the Gaussian-beam propagation to control the exponential growth of beams. In the current work, the propagator defined by the short-time asymptotic Green function allows one to construct the long time valid solution. In figure 4.2, we compare our solutions from **Algorithm 2** with the spectral solution with $\hbar = 1/256\pi$ at $t = 8$. As we increase the number of mesh points in each dimension of the computation space, our computed solution matches with the *exact* solution very well.

**4.1.3. Simple harmonic oscillator.** We first solve the two-dimensional simple harmonic oscillator problem where the potential is given by $V(x) = \frac{1}{2}\omega^2\|x\|_2^2$ with a
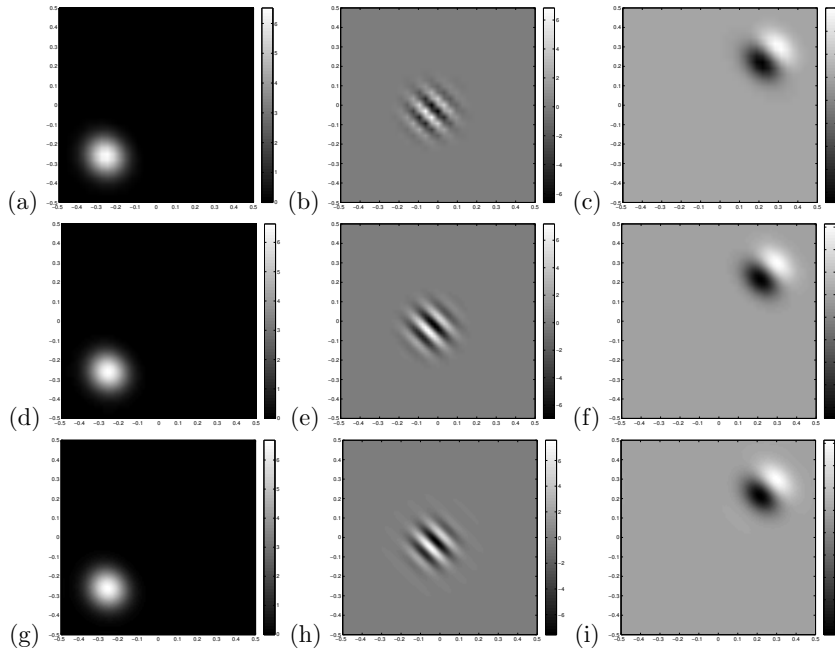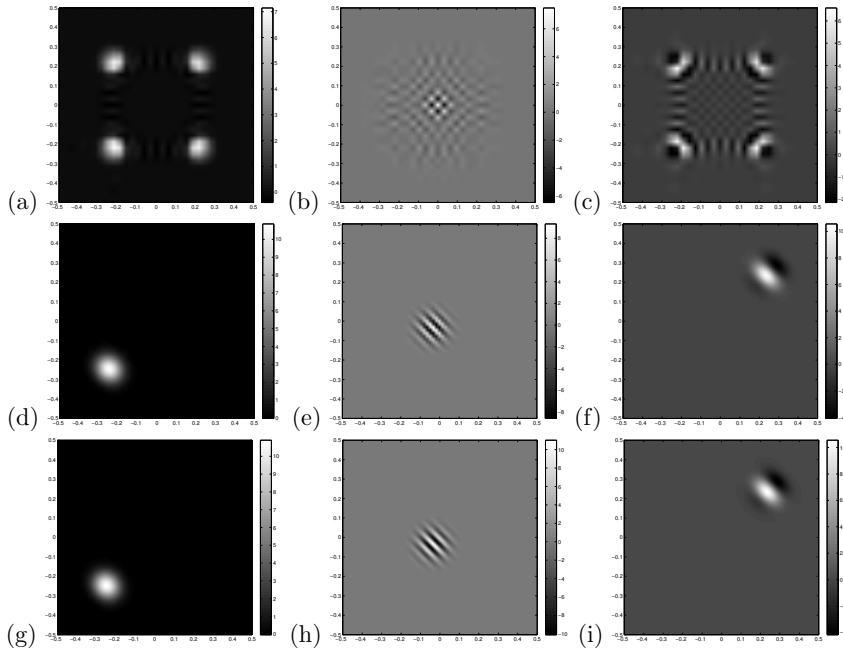
FIG. 4.3. *(Section 4.1.3 Two-Dimensional SHO with $\hbar = 1/100$,* **Algorithm 2***) Real part of the wave function at time (a) $30\Delta T$, (b) $60\Delta T$ and (c) $90\Delta T$ with $\Delta T = 0.025$, $n_x = n_\xi = 33$. Real part of the wave function at time (d) $30\Delta T$, (e) $60\Delta T$ and (f) $90\Delta T$ with $\Delta T = 0.025$, $n_x = n_\xi = 65$. Real part of the wave function at time (g) $30\Delta T$, (h) $60\Delta T$ and (i) $90\Delta T$ with $\Delta T = 0.025$, $n_x = n_\xi = 129$.*

wavepacket initial condition

$$(4.42) \qquad U_0(x) = \left(\frac{\omega}{\pi\hbar}\right)^{1/2} \exp\left(-\frac{\omega\|x - x_0\|_2^2}{2\hbar}\right) \exp\left(\frac{ip_0 \cdot x}{\hbar}\right) .$$

The natural angular frequency of oscillation $\omega$ is chosen to be 2. The packet is initially centered at the origin $(x_0 = (0,0))$ with the initial momentum $p_0 = \frac{1}{2}(-1,-1)$.

In figure 4.3 to figure 4.5, we have shown the real part of our computed solutions at various times with $\hbar = 1/100$, $\hbar = 1/200$, and $\hbar = 1/400$, respectively. For each of these figures, we compute the solution using three different sets of computational meshes, such that the coarsest solution is found on a mesh with 33 points in each dimension while the most refined solution is found on a mesh with 129 points in each dimension. As $\hbar$ decreases, we found that one has to proportionally increase the number of mesh points in order to resolve the oscillations better and obtain a more accurate solution. For comparison, we have also plotted reference solutions in figure 4.6.

We also pose the problem in the three-dimensional space and the solutions with $\hbar = 1/100$ at various times are plotted in figure 4.7.

### 4.2. Algorithm 3 and Algorithm 4 from Section 3.5.

**4.2.1. One-dimensional examples.** In this section, we consider Algorithm 3 from Section 3.5 based on the Taylor approximations in the amplitude function and
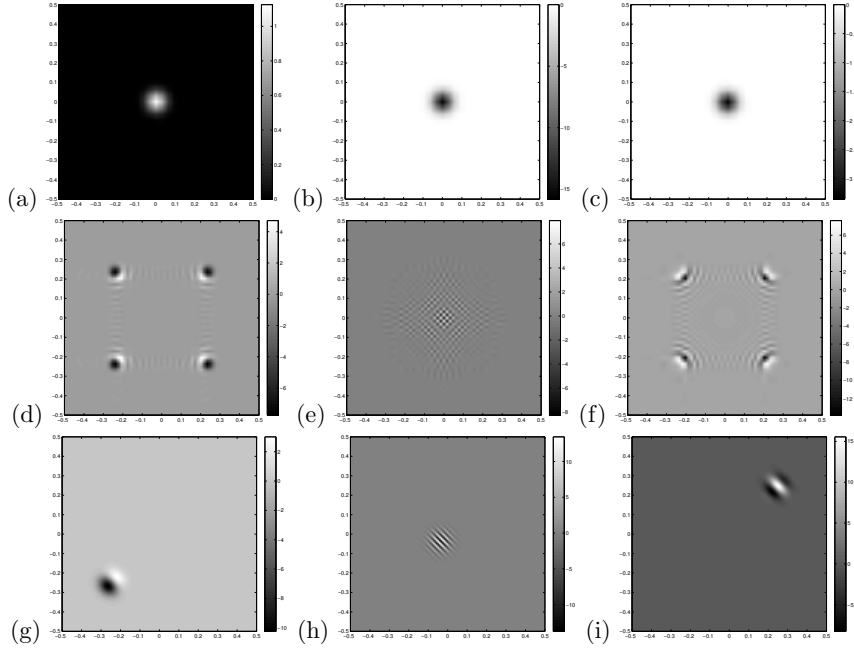
FIG. 4.4. *(Section 4.1.3 Two-Dimensional SHO with $\hbar = 1/200$, Algorithm 2) Real part of the wave function at time (a) $30\Delta T$, (b) $60\Delta T$ and (c) $90\Delta T$ with $\Delta T = 0.025$, $n_x = n_\xi = 33$. Real part of the wave function at time (d) $30\Delta T$, (e) $60\Delta T$ and (f) $90\Delta T$ with $\Delta T = 0.025$, $n_x = n_\xi = 65$. Real part of the wave function at time (g) $30\Delta T$, (h) $60\Delta T$ and (i) $90\Delta T$ with $\Delta T = 0.025$, $n_x = n_\xi = 129$.*

TABLE 4.1
*(Section 4.2) SHO solutions with $\hbar = 1/256$ using Algorithm 3.*

| $s$: Mesh $n = 2^s$ | 1-norm | Rate | 2-norm | Rate | $\infty$-norm | Rate |
|---|---|---|---|---|---|---|
| 13 | $4.41 \times 10^{-1}$ | - | $1.12 \times 10^0$ | - | $4.00 \times 10^0$ | - |
| 14 | $1.17 \times 10^{-1}$ | 1.91 | $2.97 \times 10^{-1}$ | 1.91 | $1.06 \times 10^0$ | 1.92 |
| 15 | $2.94 \times 10^{-2}$ | 1.99 | $7.45 \times 10^{-2}$ | 2.00 | $2.67 \times 10^{-1}$ | 1.99 |
| 16 | $7.37 \times 10^{-3}$ | 2.00 | $1.87 \times 10^{-2}$ | 1.99 | $6.68 \times 10^{-2}$ | 2.00 |
| 17 | $1.84 \times 10^{-3}$ | 2.00 | $4.67 \times 10^{-3}$ | 2.00 | $1.67 \times 10^{-2}$ | 2.00 |
| 18 | $4.61 \times 10^{-4}$ | 2.00 | $1.12 \times 10^{-3}$ | 2.06 | $4.18 \times 10^{-3}$ | 2.00 |
| 19 | $1.15 \times 10^{-4}$ | 2.00 | $2.92 \times 10^{-4}$ | 1.94 | $1.04 \times 10^{-3}$ | 2.01 |
| 20 | $2.89 \times 10^{-5}$ | 1.99 | $7.29 \times 10^{-5}$ | 2.00 | $2.61 \times 10^{-4}$ | 1.99 |

the phase function. We first consider the simple one-dimensional Harmonic oscillator with $\hbar = 1/256$ and we compute the solution for $t = 2\pi$. We compare our solutions with the exact solution and Table 4.1 shows the errors in the wave function using different norm measurements in Table 4.1, and the approximation converges to the exact solution in the second order.

We have also repeated the computations for the caustic example from section 4.1.1 and we have shown the solution in figure 4.8. It is worth noticing that we are using only $n = 2^{12} = 4096$ points in the computations, and this number is significantly less than the total number of grid points that Algorithm 2 used for the same example

FIG. 4.5. *(Section 4.1.3 Two-Dimensional SHO with $\hbar = 1/400$,* **Algorithm 2***) Real part of the wave function at time (a) $30\Delta T$, (b) $60\Delta T$ and (c) $90\Delta T$ with $\Delta T = 0.025$, $n_x = n_\xi = 33$. Real part of the wave function at time (d) $30\Delta T$, (e) $60\Delta T$ and (f) $90\Delta T$ with $\Delta T = 0.025$, $n_x = n_\xi = 65$. Real part of the wave function at time (g) $30\Delta T$, (h) $60\Delta T$ and (i) $90\Delta T$ with $\Delta T = 0.025$, $n_x = n_\xi = 129$.*
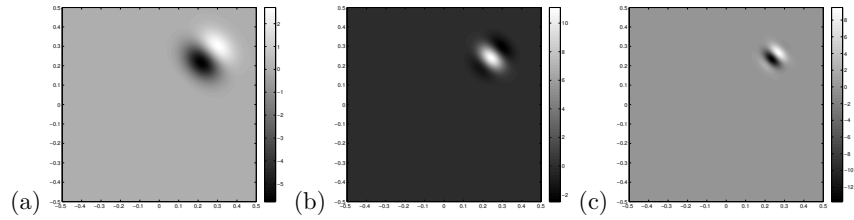


FIG. 4.6. *(Section 4.1.3 Two-Dimensional SHO) Reference solutions at $t = 2.25$ for (a) $\hbar = 1/100$, (b) $\hbar = 1/200$ and (c) $\hbar = 1/400$.*
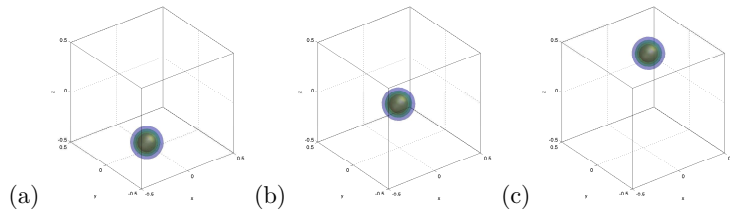


FIG. 4.7. *(Section 4.1.3 Three-Dimensional SHO with $\hbar = 1/100$,* **Algorithm 2***) Position density at time (a) $30\Delta T$, (b) $60\Delta T$ and (c) $90\Delta T$ with $\Delta T = 0.025$, $n_x = n_\xi = 33$.*
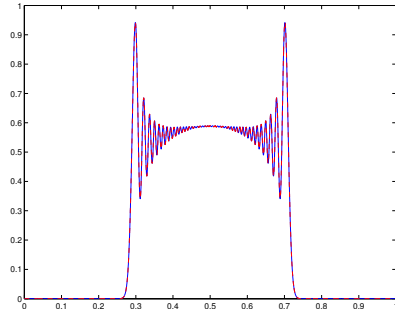
FIG. 4.8. *(Section 4.2) The one-dimensional example with caustics.* $\hbar = 1/1024$ *with* $n = 2^{12}$ *using Algorithm 3.*
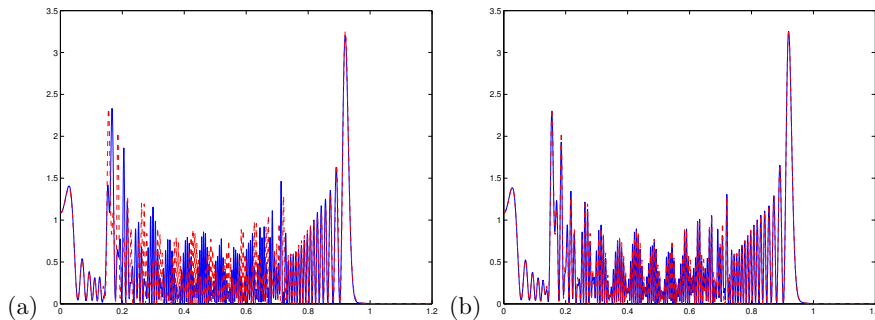


FIG. 4.9. *(Section 4.2) Solutions under the cosine potential with* $\hbar = 1/256\pi$ *with (a)* $n = 2^{17}$ *and (d)* $n = 2^{20}$. *The red-dashed curves are the intensities obtained by the spectral method, while the blue-solid lines are computed using the proposed algorithm Algorithm 3.*

with $n_x = n_\xi = 1025$.

Figures 4.9 and 4.10 show the solutions for the cosine potential example that we used in section 4.1.2. In figure 4.10, we plot the solution using various mesh points in discretizing the physical space. Indeed, in order to obtain a reasonably accurate solution, one needs to use as many as $n = 2^{20} = 1048576$ points in the approximation, figure 4.10(d). However, the number is still smaller than $4096^2 = 2^{24}$, which is needed in obtaining the solution in figure 4.2(d). Moreover, since the computational complexity for the algorithm is low, the total CPU time required is less than 4 minutes.

We have applied the higher-order Taylor expansion based approximation to the same example. The solutions are shown in figure 4.11. The phase is correctly computed when $n = 2^{18}$ as shown in figure 4.11 (b). And the computed solution is almost indistinguishable from the reference solution when we further double the number of mesh points. Concerning the extra computation effort, the total CPU time for obtaining the solutions are 4.57s, 20.28s, 87.66s and 358.20s for $n = 2^{17}$, $2^{18}$, $2^{19}$, and $2^{20}$, respectively.

To further check with the computational complexity, we have plotted the CPU times in figure 4.12. To determine the solution at a fixed final time, the number of time marching steps is inversely proportional to $\Delta x$, and therefore is proportional to $n$. Therefore, the total computational complexity for the whole algorithm to reach
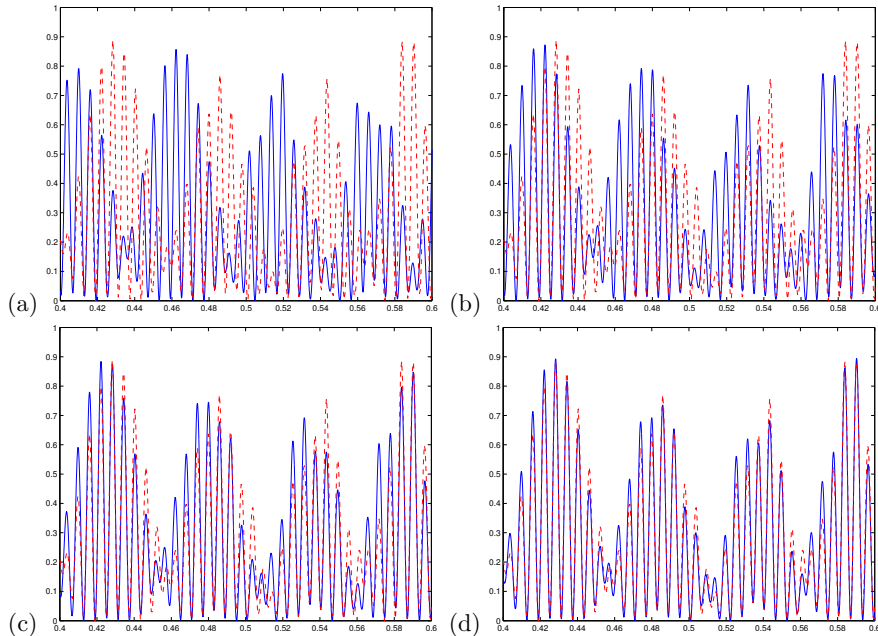
FIG. 4.10. *(Section 4.2) Zoom-in solutions under the cosine potential with $\hbar = 1/256\pi$ with (a) $n = 2^{17}$, (b) $n = 2^{18}$, (c) $n = 2^{19}$ and (d) $n = 2^{20}$. Solutions are computed in (a) 2.35s, (b) 10.94s, (c) 49.42s and (d) 207.51s. The number of iterations are (a) 122, (b) 244, (c) 487 and (d) 973. The red-dashed curves are the intensities obtained by the spectral method, while the blue-solid lines are computed using the proposed algorithm* **Algorithm 3***.*

the final time is $O(n^2 \log n)$. In figure 4.12, we have collected the CPU running time for the above example using both **Algorithm 3** (in red circles) and **Algorithm 4** (in red squares) and have drawn the fitted lines in solid. We found that the total CPU running time matches very well with the expected $n^2 \log n$ trend, where the total number of time steps is proportional to $O(n)$.

In Figure 4.13, we have demonstrated several convergence tests for both the low-order Taylor scheme **Algorithm 3** (3.30) and the high-order Taylor approximation **Algorithm 4** (3.35). In figures 4.13(a)(b), we show the errors in the quantity $\int |U(x,t)|^2 dx$ for various meshes under the same cosine potential with $\hbar = 1/256\pi$. As demonstrated in Section 3.5.2, the low-order scheme (3.30) preserves the value up to the machine epsilon, shown in figure 4.13(a). In figure 4.13(b), we found that the error increases in time. As we double the number of mesh points, the error at a fixed time is approximately halved. To check with the error bound in Theorem 3.1, we plot in figure 4.13(c) the error in $\int |U(x, \Delta t)|^2 dx$ for various $\Delta t$ using a fixed number of mesh points $n = 2^{18}$. The slope of the least-squares fitting of the log-log data is 4.009 which matches well with the theoretical prediction $O(\Delta t^4)$. In figure 4.13(d), we consider the infinity-norm error in the intensities between the time-splitting spectral solution and our numerical solutions from **Algorithm 3** and **Algorithm 4**, respectively, i.e.

$$E_{\Delta x} = \max_x |I_{\text{exact}}(x) - I_{\Delta x}(x)| .$$

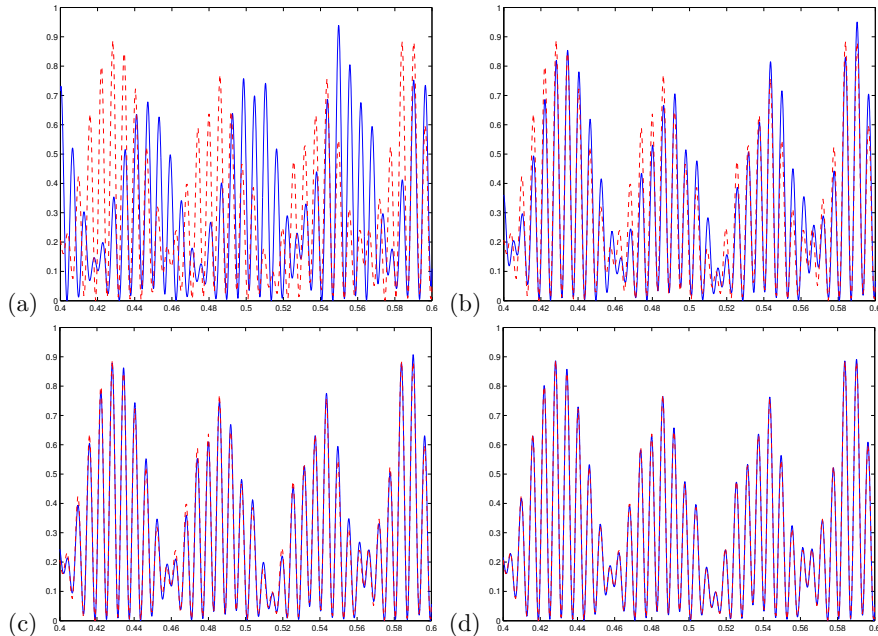The solutions from our high-order scheme are in general more accurate than those

FIG. 4.11. *(Section 4.2) Zoom-in solutions under the cosine potential with $\hbar = 1/256\pi$ using the higher order Green's function approximation (3.33) with (a) $n = 2^{17}$, (b) $n = 2^{18}$, (c) $n = 2^{19}$ and (d) $n = 2^{20}$. The CPU times are 4.57s, 20.28s, 87.66s and 358.20s, respectively. The number of iterations are (a) 122, (b) 244, (c) 487 and (d) 973. The red-dashed curves are the intensities obtained by the spectral method, while the blue-solid lines are computed using the proposed algorithm Algorithm 4.*

from the low-order version with the rate of convergence to $U_{\text{exact}}$ approximately of order two.

**4.2.2. Two-dimensional cosine potential.** In this example, we consider the evolution of a Gaussian wave function

$$(4.43) \qquad\qquad U(r,0) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[ -\frac{x^2 + y^2}{4\sigma^2} \right]$$

under a cosine potential

$$V(x,y) = \frac{1}{8}\cos(2\pi x)\cos(2\pi y)$$

with $\sigma = 0.1$ and $\hbar = 1/32\pi$. The number of mesh points along each direction is $n = 2048$ and we are looking for the solution at $t = 4$. Figures 4.14 and figure 4.15 show the solutions using the low-order Taylor-expansion based Green function (3.29) and the high-order one (3.33), respectively. The complicated wave structure can be observed in figure 4.14 and figure 4.15 (c-f), where we have plotted the real part of the wave function at $t = 1$ to $t = 4$. To better distinguish the solutions, we also plot the cross-sections of these solutions in figure 4.16.

**4.2.3. Two-dimensional and three-dimensional Gaussian potentials.** In the next example, we consider another complicated two-dimensional case where the
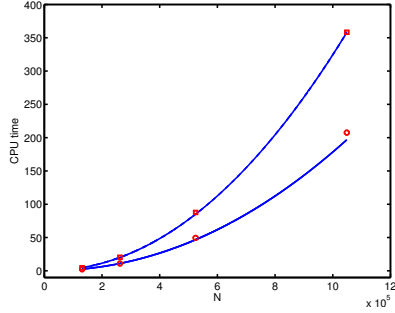
FIG. 4.12. *(Section 4.2) Computational complexities of **Algorithm 3** (red circles) and **Algorithm 4** (red squares). The solid line is the fitting line for $n^2 \log n$ which is the overall complexity including the total number of time steps, where $N = n$ in this one-dimensional example, the cost for each time step is $O(n \log n)$, and the total number of time step is $O(n)$.*
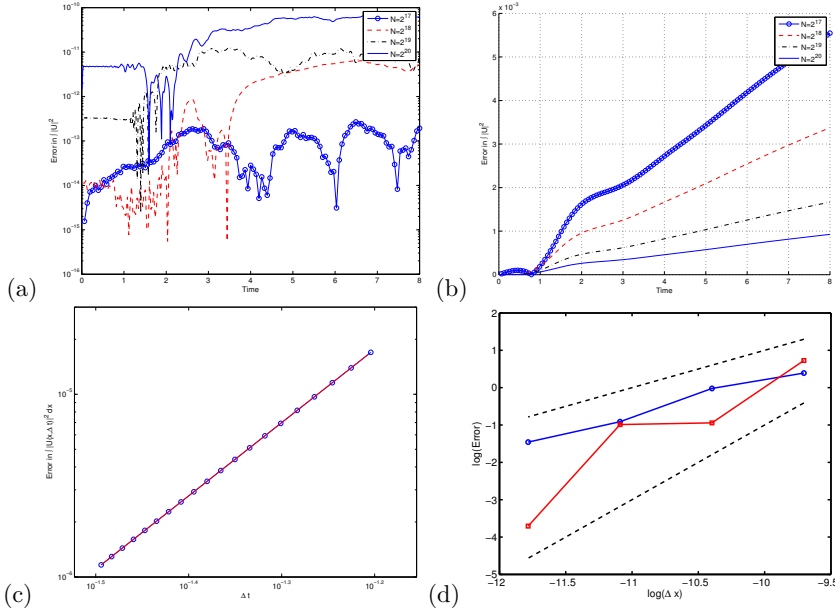


FIG. 4.13. *(Section 4.2) Numerical error in $\int |U(x,t)|^2 dx$ under the cosine potential with $\hbar = 1/256\pi$ using (a) the low order Taylor approximation scheme **Algorithm 3** (3.30) and (b) the high order Taylor approximation scheme **Algorithm 4** (3.35) for various n. (c) Numerical error in $\int |U(x,\Delta t)|^2 dx$ for various $\Delta t$ with $n = 2^{18}$ using **Algorithm 4**. The slope of the log-log graph is 4.009. (d) Numerical error in $\int |U(x,t_f)|^2 dx$ with $t_f = 8$ for various n using **Algorithm 3** (blue circles) and **Algorithm 4** (red squares). The slope of the upper- and the lower- dashed lines are 1 and 2, respectively.*

potential has a Gaussian hump given by

$$V(x,y) = \exp\left[-\frac{x^2 + y^2}{2\sigma_v^2}\right],$$
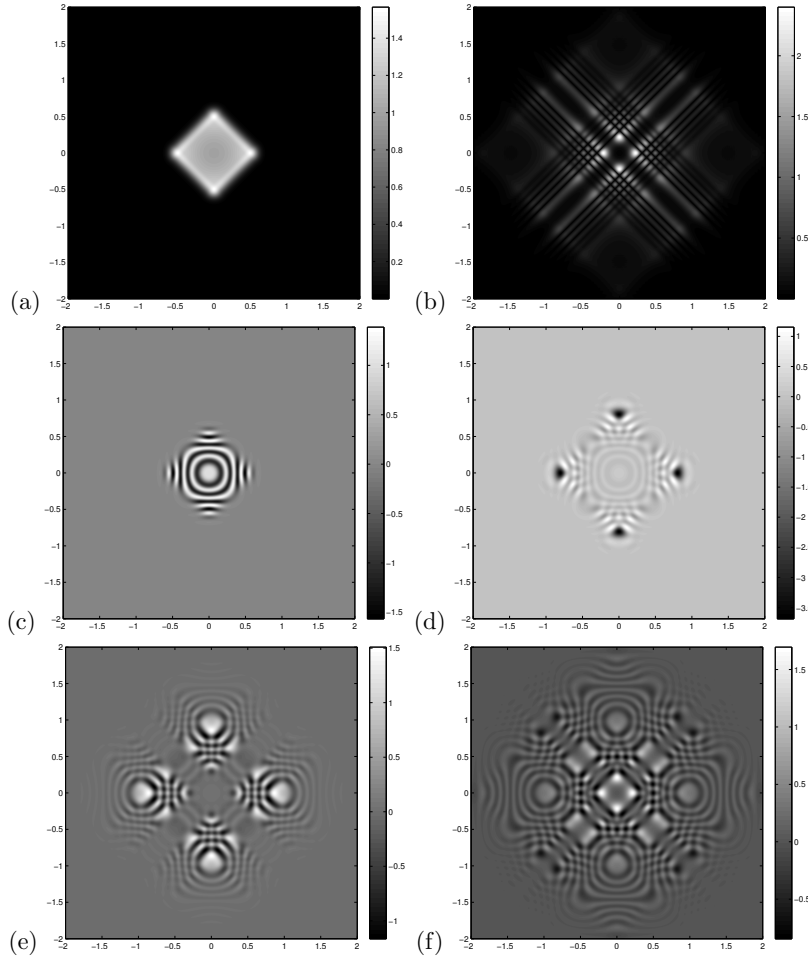
FIG. 4.14. *(Section 4.2) Two-dimensional example using the approximated propagator Algorithm 3. Cosine potential with initial Gaussian wavepacket centered at the origin $\hbar = 1/32\pi$. Intensities using $n = 2048$ at (a) 1.0 and (b) 4.0. Real part of the wave function at (c) 1.0, (d) 2.0, (e) 3.0 and (f) 4.0. The total CPU time is approximately 31s.*

with $\sigma_v = 0.2$. The initial condition is given by a Gaussian wave packet with the standard deviation $\sigma = 0.2$, momentum $p_0 = (1,1)$ and mean position centered at $(x_0, y_0) = (-1.5, -1.5)$. We compute the solutions up to $t = 2$. In figure 4.17 we have plotted some ray trajectories under the potential from locations within the support of the initial Gaussian packet, where the initial location of each ray is plotted using red circle and the trajectories are shown in solid blue lines. Because of the Gaussian barrier near the origin, caustics occur which can be clearly observed from the rays. Figure 4.18 shows the numerical solutions using the approximated propagator (3.29). We have also applied high-order Taylor approximation (3.33) as shown in figure 4.19. To check the accuracy in our solutions, we compare them with the numerical solution obtained by the time-splitting spectral method with $\Delta t = \Delta x/4$ with $n = 2048$ in
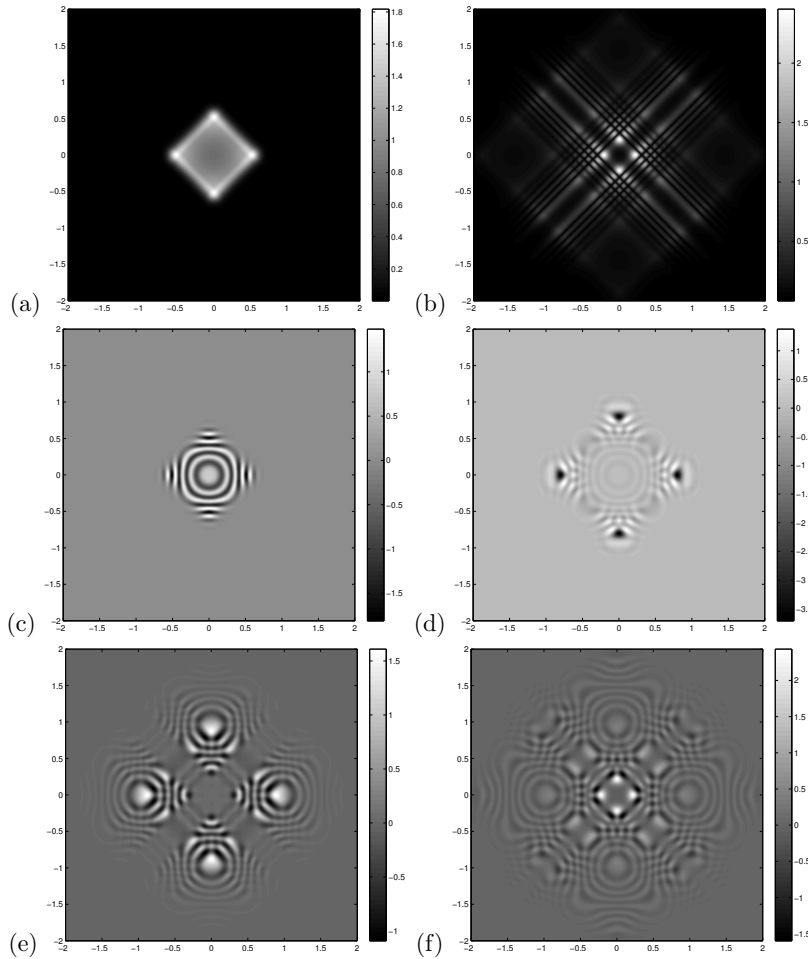
FIG. 4.15. *(Section 4.2) Two-dimensional example using the higher order Taylor expansion approximated propagator* **Algorithm 4**. *Cosine potential with initial Gaussian wavepacket centered at the origin* $\hbar = 1/32\pi$. *Intensities using* $n = 2048$ *at (a) 1.0 and (b) 4.0. Real part of the wave function using* $n = 2048$ *at (c) 1.0, (d) 2.0, (e) 3.0 and (f) 4.0. The total CPU time is approximately 56s.*

each physical direction. The real parts of these solutions along the cross section $y = 0$ are shown in figure 4.20. Note that we have gained a 20 times speedup in the computational speed using the convolution method.

We also extend the Gaussian hump example to three-dimension by having the Gaussian potential centered at the origin with the standard deviation $\sigma_v = 0.025$. The initial condition is a Gaussian wave packet centered at $(-0.5, -0.5, -0.5)$ with momentum $(1, 1, 1)$. In figure 4.21, we solve the problem with $\hbar = 1/128$ using $n = 256$.

**4.3. Sparse Algorithms from Section 3.6.** In this section, we first study the sparse-approximation based high-order Taylor expansion spectral method, **Algorithm**
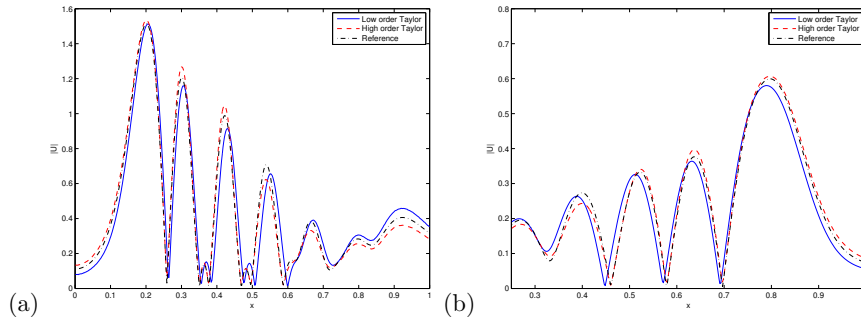
(a)  (b)

FIG. 4.16. *(Section 4.2) Two-dimensional example using the low order Taylor expansion* **Algorithm 3** *(blue solid line) and the higher order Taylor expansion approximated propagator* **Algorithm 4** *(red dashed line). Cosine potential with initial Gaussian wavepacket centered at the origin* $\hbar = 1/32\pi$. *Solutions using* $n = 2048$ *at* $t = 4.0$. *The intensities along (a)* $y = -1$ *and (b)* $y = 0$.
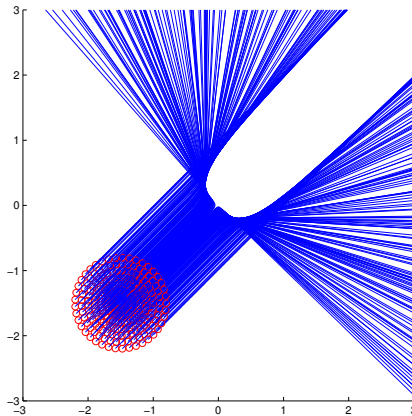


FIG. 4.17. *(Section 4.2) Ray tracing solution. Gaussian potential centered at the origin with standard deviation* $\sigma = 0.2$. *Initial Gaussian centered at* $(-1.5, -1.5)$ *with* $p_0 = (1, 1)$.

5. In figure 4.22, we have shown the final state of a two-dimensional Gaussian wave function under a Gaussian potential centered at the origin at $t = 2$. The setup is the same as in figure 4.19. Using **Algorithm 5**, we have repeated the experiment with various choices of regularization parameter $\lambda$ in the soft-thresholding. As we increase the value of $\lambda$, fewer Fourier coefficients are kept in the calculation, and more information is lost in the evolution as expected. In this example, the total number of Fourier coefficients is $2^{22}$ and we obtain an indistinguishable solution from the reference solution (using Strang's splitting spectral method in [22, 3]) by keeping only as little as 2000 of them, i.e. approximately 0.04%, for $\lambda = 10^{-3}$.

It is also straight-forward to implement the sparse-approximation based high-order Taylor expansion convolution method in **Algorithm 4'**. In figure 4.23 we have repeated the same example as in figure 4.19 with various choices of regularization parameter $\lambda$ in the soft-thresholding. In this case, the total number of Fourier coefficients is $(2 \times 2048)^2 = 2^{24}$. For $\lambda = 1$, the number of non-zero coefficients is roughly $6 \times 10^4$, which accounts for 0.36% of all coefficients. We have also checked
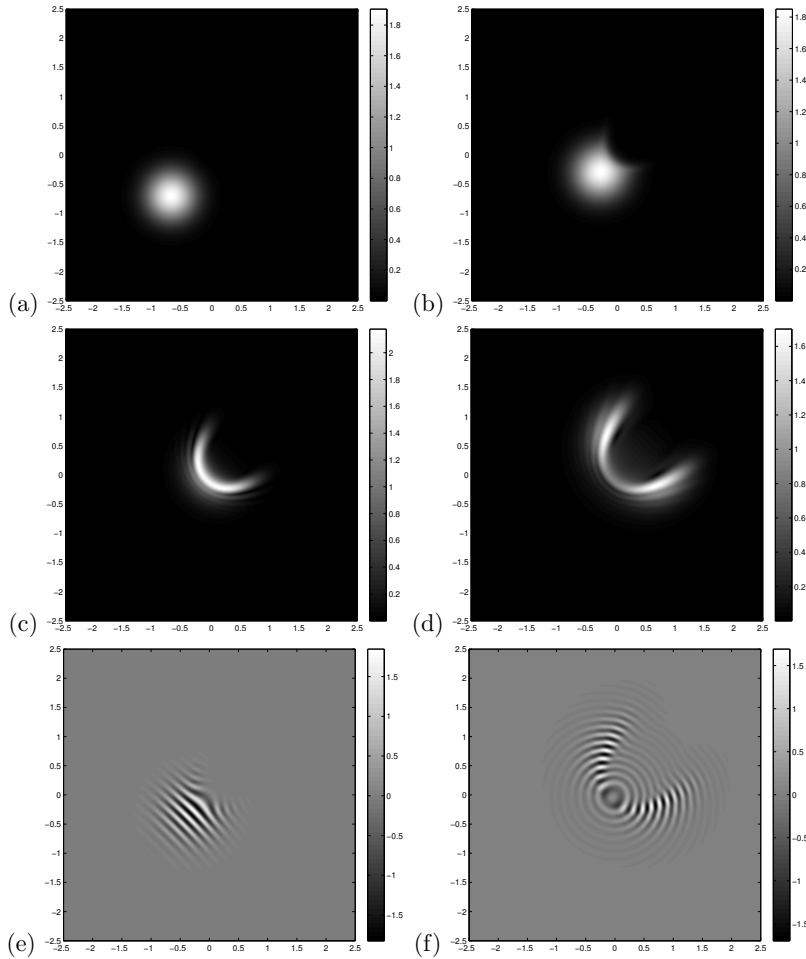
FIG. 4.18. *(Section 4.2) Two dimensional example using the approximated propagator* **Algorithm 3**. *Gaussian potential centered at the origin with standard deviation* $\sigma = 0.2$. *Initial Gaussian centered at* $(-1.5, -1.5)$ *with* $p_0 = (1, 1)$ *and* $\hbar = 1/32$. *Intensities using* $n = 2048$ *at (a) 0.8, (b) 1.2, (c) 1.6 and (d) 2.0. Real part of the wave function using* $n = 2048$ *at (e) 1.2 and (f) 2.0. The total CPU time is 34.81s.*

the computational speed by looking at the CPU times for various $\lambda$'s. For example, the running time for $\lambda = 10^{-4}$, $10^{-3}$, $10^{-2}$, $10^{-1}$ and $10^0$ are 92.31s, 65.30s, 55.06s, 52.99s and 52.09s, respectively. Comparing to the full implementation in which we need 63.38s to obtain the final solution, the sparse algorithm does further reduce the computational time *for large* $\lambda$ by looking at only those Fourier coefficients with *significant* contributions to the solution. Nevertheless, since the number of non-zero Fourier modes in the solution might grow in time, it might be difficult to define an *a priori* estimate for $\lambda$.

For the three-dimensional example, we consider again the simple harmonic oscillator using $n = 2^7$ mesh points in each physical dimension, which means that the
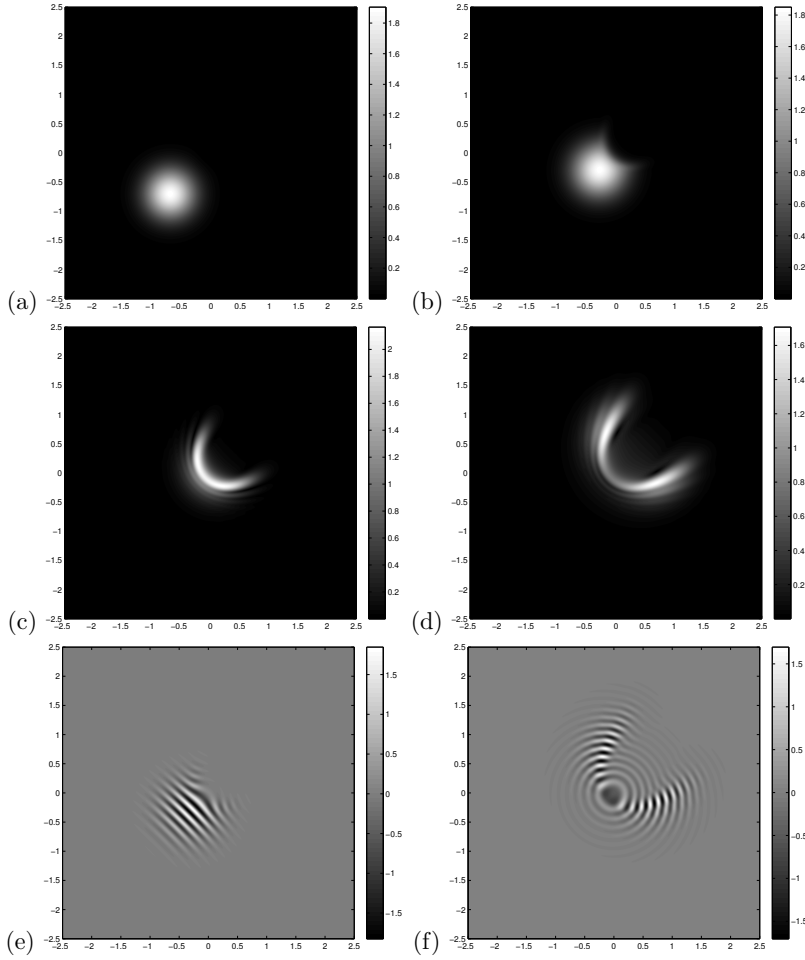
FIG. 4.19. *(Section 4.2) Two dimensional example using the higher order Taylor expansion approximated propagator* **Algorithm 4**. *Gaussian potential centered at the origin with standard deviation $\sigma = 0.2$. Initial Gaussian centered at $(-1.5, -1.5)$ with $p_0 = (1,1)$ and $\hbar = 1/32$. Intensities using $n = 2048$ at (a) 0.8, (b) 1.2, (c) 1.6 and (d) 2.0. Real part of the wave function using $n = 2048$ at (e) 1.2 and (f) 2.0. The total CPU time is 63.38s.*

total number of Fourier coefficients in **Algorithm 4** is $(2n)^3 = 2^{24}$. The running time for $\lambda = 10^{-4}$, $10^{-3}$, $\cdots$ up to $10^2$ are 16.71s, 15.42s, 15.05s, 14.48s, 13.92s, 13.59s and 13.52s, respectively. Comparing to the full implementation which takes 20.75s, there is now a significant gain in computational speed even for $\lambda = 10^{-4}$. Figure 4.24 (a) shows the error in $\int |U|^2$. Since we are considering the low-order approximation, the physical quantity $\int |U|^2$ should remain constant. Any deviation from unity is the result of the soft-thresholding. Figure 4.24 (b) plots the number of non-zero Fourier coefficients retained in the computations for various $\lambda$'s.

Figure 4.25 shows the solutions of the three-dimensional Gaussian hump example using different $\lambda$'s in the sparse implementation of the low-order Taylor expansion
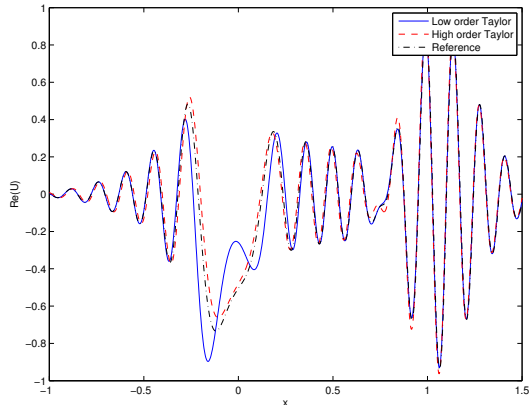
Fig. 4.20. *(Section 4.2) Two dimensional example using the low order Taylor expansion **Algorithm 3** (blue solid line) and the higher order Taylor expansion approximated propagator **Algorithm 4** (red dashed line). Gaussian potential centered at the origin with standard deviation $\sigma = 0.2$. Initial Gaussian centered at $(-1.5, -1.5)$ with $p_0 = (1, 1)$ and $\hbar = 1/32$. Solutions using $n = 2048$ at $t = 2.0$. The real part of the wave function along the cross section $y = 0$. The total CPU time for obtaining the time-splitting spectral solution is 1333.97s which is approximately 38 times and 20 times of that by **Algorithm 3** and **Algorithm 4**, respectively.*



(a)  (b)
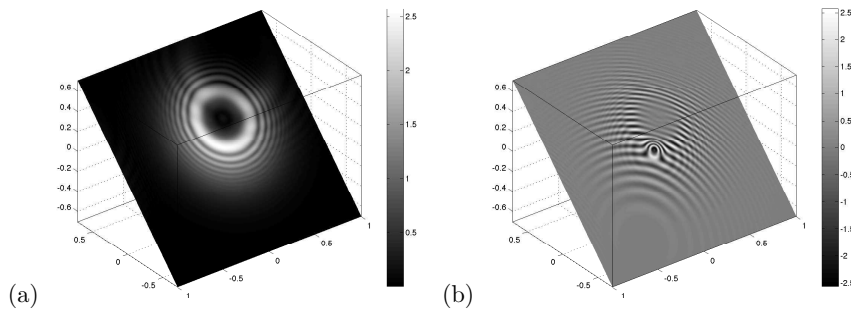
Fig. 4.21. *(Section 4.2) Three dimensional example using the approximated propagator **Algorithm 3**. Gaussian potential centered at the origin with standard deviation $\sigma = 0.05$. Initial Gaussian centered at $(-0.5, -0.5, -0.5)$ with standard deviation $\sigma = 0.02$ and $p_0 = (1, 1, 1)$. We show (a) the intensity and (b) the real part of the solution using $n = 256$ at $t = 0.6$ with $\hbar = 1/64$.*

convolution algorithm. The computational times corresponding to various $\lambda$'s are 196.28s, 279.96s, 384.40s and 667.09s for $\lambda = 10^3$, $10^2$, $10^1$ and $10^0$, respectively. For $\lambda = 10^3$, we are keeping only 3.38% of Fourier coefficients at the final time step, figure 4.26, and the solution is still quantitatively acceptable.

**4.4. Nonlinear Schrödinger equations.** In this last section, we consider some numerical solutions of the nonlinear Schrödinger equation. We consider an example from [7] to study the focusing effect in the nonlinear Schrödinger equation where the initial wave function is given by $U(x, 0) = e^{-x^2}$ and the potential is $V(x, y) = 0$. We consider $\hbar = 0.1$ and solve the solution up to $t = t_f = 1.2$. Figure 4.27 shows the computed solution using various $\Delta x$'s compared with the time-splitting spectral solution obtained by [22, 4] using $\Delta t = \Delta x/4$. We are plotting this reference solution
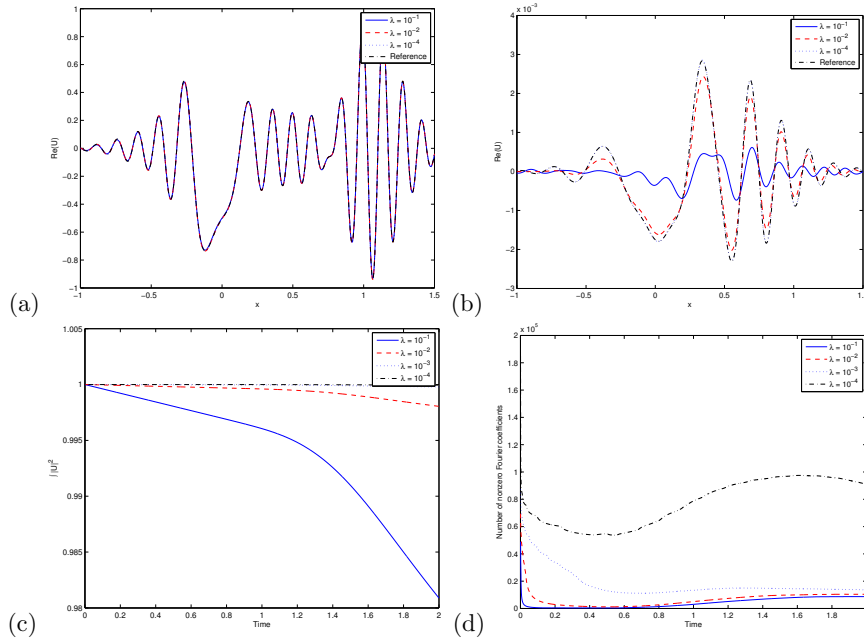
FIG. 4.22. *(Section 4.3) Two dimensional Gaussian under Gaussian potential centered at the origin with standard deviation $\sigma = 0.2$. Initial Gaussian centered at $(-1.5, -1.5)$ with $p_0 = (1, 1)$ and $\hbar = 1/32$. Solutions using $n = 2048$ at $t = 2.0$. The real part of the computed wave function using the sparse approximation higher order Taylor Expansion Spectral method* **Algorithm 5** *along the cross section (a) $y = 0$ and (b) $y = -1$ with various $\lambda$'s. (c) $\int |U(x, y, t)|^2 dxdy$ using various regularization parameters $\lambda$ and (d) J, the number of sparse coefficients for summation, using various regularization parameters $\lambda$.*

using the black dash line in figure 4.27. These solutions match pretty well. Considering the total CPU time, our numerical approach **Algorithm 6** takes approximately $9.94 \times 10^2$s, while the time-splitting spectral method takes approximately $2.52 \times 10^4$s which is more than 25 times longer than that of our approach.

**5. Conclusions.** We propose fast Huygens sweeping methods for Schrödinger equations in the semi-classical regime by incorporating short-time WKBJ propagators into Huygens' principle. Even though the WKBJ solution is valid only for a short time period due to the occurrence of caustics, Huygens' principle allows us to construct the global-in-time semi-classical solution. To improve the computational efficiency, we develop analytical approximation formulas for the short-time WKBJ propagator by using Taylor expansion in time. These analytical formulas allow us to develop families of fast Huygens sweeping methods, among which one is posed in the momentum space, and the other is posed in the position space, and both of these methods are of computational complexity $O(N \log N)$ for each time step, where $N$ is the total number of sampling points in the $d$-dimensional position space. To further speed up these methods, we also incorporate the soft-thresholding sparse strategy into our new algorithms so that the computational cost can be further reduced. The methodology can also be extended to nonlinear Schrödinger equations. One, two, and three dimensional examples demonstrate the performance of the new algorithms.
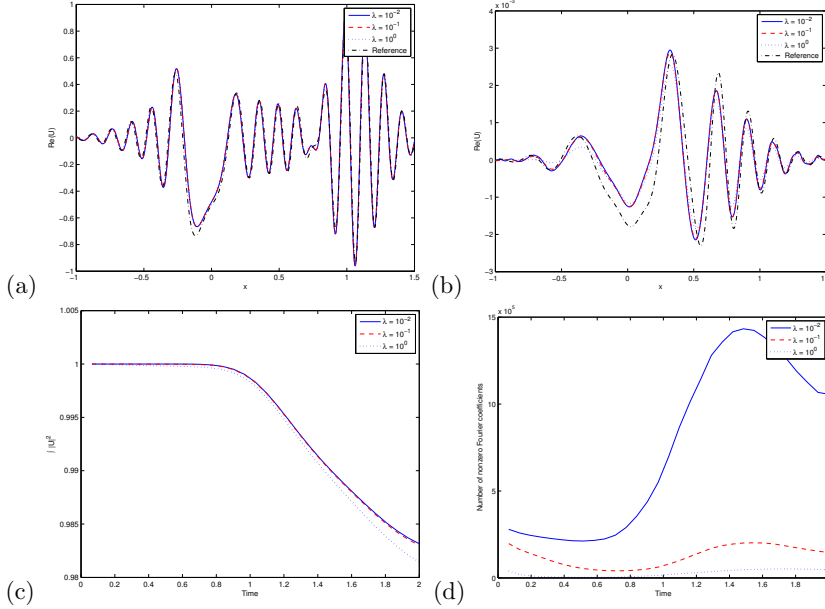
FIG. 4.23. *(Section 4.3) Two dimensional Gaussian under Gaussian potential centered at the origin with standard deviation $\sigma = 0.2$. Initial Gaussian centered at $(-1.5, -1.5)$ with $p_0 = (1, 1)$ and $\hbar = 1/32$. Solutions using $n = 2048$ at $t = 2.0$. The real part of the computed wave function using the sparse approximation higher order Taylor Expansion Convolution method Algorithm 4' along the cross section (a) $y = 0$ and (b) $y = -1$ with various $\lambda$'s. (c) $\int |U(x, y, t)|^2 dx dy$ using various regularization parameters $\lambda$ and (d) $J$, the number of sparse coefficients for summation, using various regularization parameters $\lambda$.*
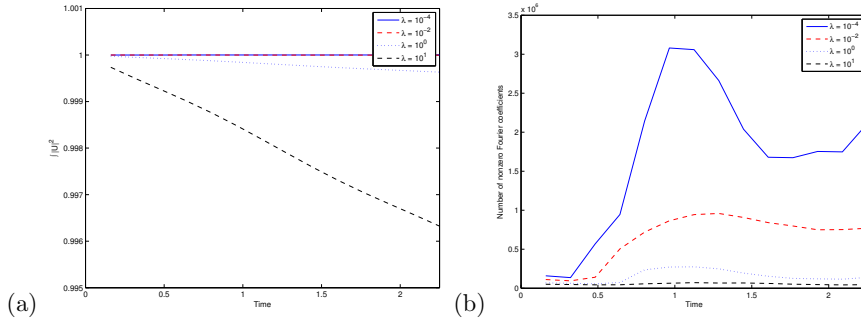


FIG. 4.24. *(Section 4.3) Three dimensional Harmonic oscillator. Solutions using the sparse approximation lower order Taylor Expansion Convolution method with $n = 2^7$ at $t = 2.25$. (a) $\int |U(x, y, t)|^2 dx dy$ using various regularization parameters $\lambda$ and (b) $J$, the number of sparse coefficients for summation, using various regularization parameters $\lambda$.*

If one is interested in the wavefunction for the linear Schrödinger equation at a single large time level $g \gg 1$, we can actually further improve the computational complexity by using the phase flow method [6, 15] in this setting. However, in the current work we are interested in the evolution of the wavefunction and we will leave the idea as a future work. Since it is very important to the community to develop
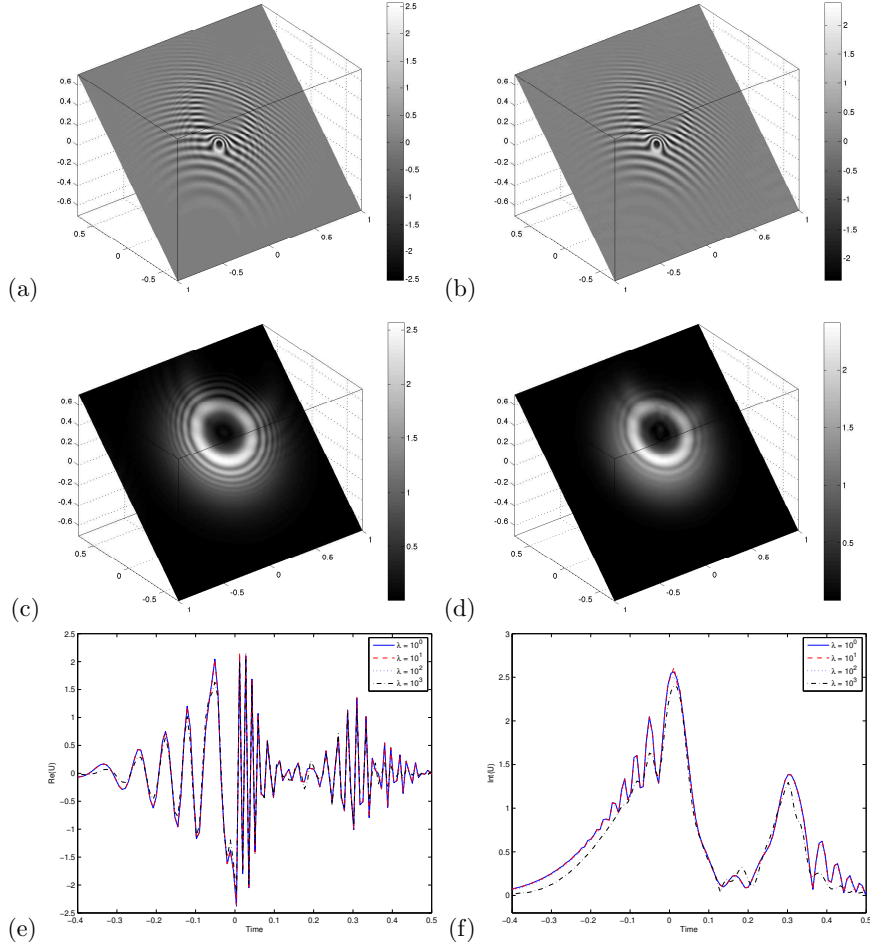
Fig. 4.25. *(Section 4.3) Three dimensional Gaussian under Gaussian potential centered at the origin with standard deviation* $\sigma = 0.2$. *Gaussian potential centered at the origin with standard deviation* $\sigma = 0.05$. *Initial Gaussian centered at* $(-0.5, -0.5, -0.5)$ *with standard deviation* $\sigma = 0.02$ *and* $p_0 = (1, 1, 1)$. *Solutions using* $n = 256$ *at* $t = 0.6$ *with* $\hbar = 1/64$. *(a) Real part of the solution with* $\lambda = 10^2$. *(b) Real part of the solution with* $\lambda = 10^3$. *(c) Position density of the solution with* $\lambda = 10^2$. *(d) Position density of the solution with* $\lambda = 10^3$. *(e) Cross section of the real part of the solution along the direction* $(1, 1, 1)$ *for various* $\lambda$'s. *(f) Cross section of the position density along the direction* $(1, 1, 1)$ *for various* $\lambda$'s.

more accurate numerical methods for nonlinear Schrödinger equation, we propose to continue extending our method in Section 3.7 to accomplish that.
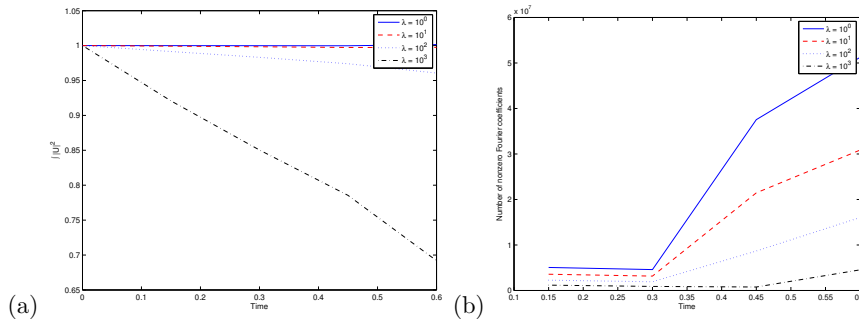
Fig. 4.26. *(Section 4.3) Three dimensional Gaussian under Gaussian potential centered at the origin with standard deviation $\sigma = 0.2$. Solutions using the sparse approximation lower order Taylor Expansion Convolution method with $n = 2^8$ at $t = 0.6$. (a) $\int |U(x,y,t)|^2 dxdy$ using various regularization parameters $\lambda$ and (b) J, the number of sparse coefficients for summation, using various regularization parameters $\lambda$.*
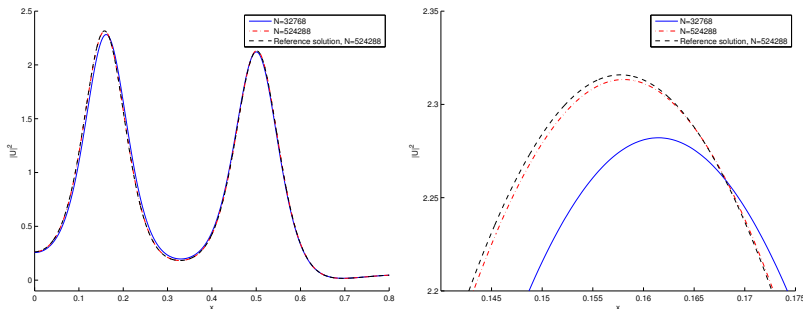


Fig. 4.27. *(Section 4.4) Nonlinear Schrödinger equation with $\hbar = 0.1$ and a zoom-in of the solution at $t = 1.2$. The total CPU times using our method **Algorithm 6** and using the spectral method in [22, 4] with $n = 2^{19}$ are approximately 16 minutes and 420 minutes, respectively.*

## REFERENCES

[1]  V. M. BABICH AND V. S. BULDYREV, *Asymptotic Methods in Short Wave Diffraction Problems (in Russian)*, Nauka, Moscow, 1972.

[2]  G. BAO, J. QIAN, L. YING, AND H. ZHANG, *A convergent multiscale Gaussian-beam parametrix for the wave equation*, Commun. PDEs, 38 (2013), pp. 92–134.

[3]  W. BAO, S. JIN, AND P. A. MARKOWICH, *On time-splitting spectral approximations for the schrödinger equation in the semiclassical regime*, J. Comput. Phys., 175 (2002), pp. 487–524.

[4]  W. BAO, S. JIN, AND P. A. MARKOWICH, *Numerical study of time-splitting spectral discretizations of nonlinear Schrödinger equations in the semiclassical regimes*, SIAM J. Sci. Comput., 25 (2003), pp. 27–64.

[5]  S. BOUGACHA, J. AKIAN, AND R. ALEXANDRE, *Gaussian beams summation for the wave equation in a convex domain*, Commun. Math. Sci., 7 (2009), pp. 973–1008.

[6]  E. J. CANDÈS AND L. YING, *Fast geodesics computation with the phase flow method*, J. Comput. Phys., 220 (2006), pp. 6–18.

[7]  H. D. CENICEROS AND F.-R. TIAN, *A numerical study of the semi-classical limit of the focusing nonlinear Schrödinger equation*, Phys. Lett. A, 306 (2002), pp. 25–34.

[8]  S. FOMEL, L. YING, AND X. SONG, *Seismic wave extrapolation using lowrank symbol approximation*, Geophysical Prospecting, 2012.

[9]  E. J. HELLER, *Cellular dynamics: A new semiclassical approach to time-dependent quantum*

*mechanics*, J. Chem. Phys., 94 (1991), pp. 2723–2729.

[10] E. J. Heller, *Guided gaussian wave packets*, Acc. Chem. Res., 39 (2006), pp. 127–134.

[11] N. R. Hill, *Gaussian beam migration*, Geophysics, 55 (1990), pp. 1416–1428.

[12] G. S. Jiang and D. Peng, *Weighted ENO schemes for Hamilton-Jacobi equations*, SIAM J. Sci. Comput., 21 (2000), pp. 2126–2143.

[13] E. Kluk, M. F. Herman, and H. L. Davis, *Comparison of the propagation of semiclassical frozen gaussian wave functions with quantum propagation for a highly excited anharmonic oscillator*, J. Chem. Phys., 84 (1986), pp. 326–334.

[14] S. Leung and J. Qian, *Eulerian Gaussian beams for Schrödinger equations in the semiclassical regime*, J. Comput. Phys., 228 (2009), pp. 2951–2977.

[15] S. Leung and J. Qian, *The backward phase flow and FBI-transform-based Eulerian Gaussian beams for the Schrödinger equation*, J. Comput. Phys., 229 (2010), pp. 8888–8917.

[16] S. Leung, J. Qian, and R. Burridge, *Eulerian Gaussian beams for high frequency wave propagation*, Geophysics, 72 (2007), pp. SM61–SM76.

[17] S. Leung and H. K. Zhao, *Gaussian beam summation for diffraction in inhomogeneous media based on the grid based particle method*, Communications in Computational Physics, 8 (2010), pp. 758–796.

[18] S. Luo, J. Qian, and R. Burridge, *Fast Huygens sweeping methods for Helmholtz equations in inhomogeneous media*, submitted:xxx, 2012.

[19] S. Luo, J. Qian, and R. Burridge, *High-order factorization based high-order fast sweeping methods for point-source eikonal equations*, SIAM J. Numer. Analy., 52 (2014), pp. 23–44.

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

[21] S. J. Osher and C. W. Shu, *High-order Essentially NonOscillatory schemes for Hamilton-Jacobi equations*, SIAM J. Num. Anal., 28 (1991), pp. 907–922.

[22] D. Pathria and J. Morris, *Pseudo-spectral solution of nonlinear Schrödinger equations*, J. Comput. Phys., 87 (1990), pp. 108–125.

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

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

[25] J. Qian and L. Ying, *Fast multiscale Gaussian wavepacket transforms and multiscale Gaussian beams for the wave equation*, SIAM J. Multi. Model. & Simul., 8 (2010), pp. 1803–1837.

[26] J. Ralston, *Gaussian beams and the propagation of singularities*, Studies in partial differential equations, 23 (1983), pp. 206–248.

[27] H. Schaeffer, S. Osher, R. Caflisch, and C. Hauck, *Sparse dynamics for partial differential equations*, UCLA CAM Report (12-74), 2012.

[28] L. S. Schulman, *Techniques and Applications of Path Integration*, Wiley, New York, 1981.

[29] S. Serna and J. Qian, *Fifth order weighted power-ENO methods for Hamilton-Jacobi equations*, J. Sci. Comput., 29 (2006), pp. 57–81.

[30] N. Tanushev, B. Engquist, and R. Tsai, *Gaussian beam decomposition of high frequency wave fields*, J. Comput. Phys., 228 (2009), pp. 8856–8871.

[31] N. Tanushev, J. Qian, and J. Ralston, *Mountain waves and gaussian beams*, SIAM J. Multi. Model. & Simul., 6 (2007), pp. 688–709.