Semi-Eulerian and High Order Gaussian Beam Methods for the Schrödinger Equation in the Semiclassical Regime [*]

Shi Jin[†], Hao Wu[‡], and Xu Yang[§]

October 9, 2009

Abstract

A novel Eulerian Gaussian beam method was developed in [8] to compute the Schrödinger equation efficiently in the semiclassical regime. In this paper, we introduce an efficient semi-Eulerian implementation of this method. The new algorithm inherits the essence of the Eulerian Gaussian beam method where the Hessian is computed through the derivatives of the complexified level set functions instead of solving the dynamic ray tracing equation. The difference lies in that, we solve the ray tracing equations to determine the centers of the beams and then compute quantities of interests only around these centers. This yields effectively a local level set implementation, and the beam summation can be carried out on the initial physical space instead of the phase plane. As a consequence, it reduces the computational cost and also avoids the delicate issue of beam summation around the caustics in the Eulerian Gaussian beam method. Moreover, the semi-Eulerian Gaussian beam method can be easily generalized to higher order Gaussian beam methods, which is the topic of the second part of this paper. Several numerical examples are provided to verify the accuracy and efficiency of both the first order and higher order semi-Eulerian methods.

Key words. Schrödinger equation, semi-Eulerian Gaussian beam method, high order methods

AMS subject classifications. 81Q20, 65M99

[†]Department of Mathematics, University of Wisconsin, Madison, WI 53706, USA (jin@math.wisc.edu)

[‡]Department of Mathematical Sciences, Tsinghua University, Beijing, 10084, China (hwu@tsinghua.edu.cn)

[§]Program in Applied and Computational Mathematics, Princeton University, NJ 08544, USA (xuyang@math.princeton.edu)

1

1 Introduction

The Schrödinger equation is the fundamental equation in quantum mechanics. The rescaled linear Schrödinger equation can be written as

$$i\varepsilon\frac{\partial \Psi^\varepsilon}{\partial t} + \frac{\varepsilon^2}{2}\Delta\Psi^\varepsilon - V(\boldsymbol{x})\Psi^\varepsilon = 0, \quad \boldsymbol{x} \in \mathbb{R}^n, \tag{1.1}$$

where $\Psi^\varepsilon(\boldsymbol{x}, t)$ is the wave function, $V(\boldsymbol{x})$ is the potential, ε is the re-scaled Plank constant that describes the ratio between quantum time/space scale and the macroscopic time/space scale. This scaling corresponds to the so-called semiclassical regime. In this paper we consider (1.1) with the WKB-initial condition

$$\Psi^\varepsilon(0, \boldsymbol{x}) = A_0(\boldsymbol{x})e^{iS_0(\boldsymbol{x})/\varepsilon}. \tag{1.2}$$

The direct numerical simulation of (1.1)-(1.2) has the difficulty that when ε is small the wave function $\Psi^\varepsilon(\mathbf{x}, t)$ becomes oscillatory of wave length $O(\varepsilon)$. The best direct numerical solver so far is the time splitting spectral method which requires a mesh size of $O(\varepsilon)$ [1]. Gaussian beam methods are asymptotic methods for such high frequency waves which allow numerical meshes to be $O(\sqrt{\varepsilon})$, and they outperform the classical geometric optics method in that the Gaussian beam approximations are accurate even around caustics. While the classical Gaussian beam methods are in the Lagrangian framework [2, 3, 4, 5, 6, 16, 17], there have been very recent efforts in developing Eulerian Gaussian beam methods [12, 8, 11]. The error analysis on these Eulerian methods and their higher order extension were performed in [13, 14].

For more recent works in Guassian beam methods the readers are also referred to [20, 19, 15].

We first summarize the Eulerian Gaussian beam method proposed in [8]. Consider the ansatz

$$\varphi_{eu}^\varepsilon(t, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{p}) = A(t, \boldsymbol{y}, \boldsymbol{p})e^{iT(t, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{p})/\varepsilon}, \tag{1.3}$$

where

$$T(t, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{p}) = S(t, \boldsymbol{y}, \boldsymbol{p}) + \boldsymbol{p} \cdot (\boldsymbol{x} - \boldsymbol{y}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{y})^\top M(t, \boldsymbol{y}, \boldsymbol{p})(\boldsymbol{x} - \boldsymbol{y}).$$

Here $(\boldsymbol{y}, \boldsymbol{p})$ is defined by the following Hamiltonian system

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = \boldsymbol{p}, \tag{1.4}$$

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla_{\boldsymbol{y}}V, \tag{1.5}$$

Around each ray $\boldsymbol{y}$ a Gaussian profile of form (1.3) is constructed, with a width of $O(\sqrt{\varepsilon})$. The asymptotic wave function is constructed via the

following Eulerian Gaussian beam summation over all beams:

$$\Phi_{eu}^{\varepsilon}(t,\boldsymbol{x}) = \int_{\mathbb{R}^n}\int_{\mathbb{R}^n}\left(\frac{1}{2\pi\varepsilon}\right)^{\frac{n}{2}} r_{\theta}(\boldsymbol{x}-\boldsymbol{y})\varphi_{eu}^{\varepsilon}(t,\boldsymbol{x},\boldsymbol{y},\boldsymbol{p})\Pi_{j=1}^{n}\delta(\mathrm{Re}[\phi_j])\mathrm{d}\boldsymbol{p}\mathrm{d}\boldsymbol{y},$$

(1.6)

where $\boldsymbol{\phi}=(\phi_1,\cdots,\phi_n)$ satisfies the Liouville equations

$$\mathcal{L} = \partial_t + \boldsymbol{p}\cdot\nabla_{\boldsymbol{y}} - \nabla_{\boldsymbol{y}}V\cdot\nabla_{\boldsymbol{p}}, \tag{1.7}$$
$$\mathcal{L}\boldsymbol{\phi} = 0, \tag{1.8}$$

with the *complex* initial data

$$\boldsymbol{\phi}_0(\boldsymbol{y},\boldsymbol{p}) = -i\boldsymbol{y} + (\boldsymbol{p}-\nabla_{\boldsymbol{y}}S_0). \tag{1.9}$$

The Hessian M is obtained by

$$M = -(\nabla_{\boldsymbol{p}}\boldsymbol{\phi})^{-1}\nabla_{\boldsymbol{y}}\boldsymbol{\phi}. \tag{1.10}$$

The physical quantities $\boldsymbol{u}=\nabla_{\boldsymbol{y}}S$, S and A are given by [8, 11]

$$\phi_j(t,\boldsymbol{y},\boldsymbol{p}) = 0, \quad \text{at} \quad \boldsymbol{p}=\boldsymbol{u}(t,\boldsymbol{y}), \quad j=1,\cdots,n,$$
$$\mathcal{L}S = \frac{1}{2}|\boldsymbol{p}|^2 - V,$$
$$\mathcal{L}A = \frac{1}{2}\mathrm{Tr}(MA).$$

This Eulerian formulation differs from the classical Lagrangian method in two aspects:

1. The Hessian (1.10) is computed through the derivatives of the complex-valued level set functions instead of the Riccati equation or the dynamic ray tracing equations. This greatly reduces the number of equations to be solved and consequently simplifies significantly the computational complexity and cost.

2. The Eulerian summation integral (1.6) sits in the $\boldsymbol{y}-\boldsymbol{p}$ plane, while the traditional Lagrangian one is in the $\boldsymbol{y}_0$–the initial beam position–plane. As pointed out in [8], the numerical discretization of (1.6) around caustics needs some special care. When there exists a large number of caustics, the implement will be either technically nontrivial or need to introduce some numerical dissipation by discretizing the delta function.

In this paper we develop a semi-Eulerian Gaussian beam method which *inherits* the good property 1 and *improves* the implementation in point 2 of the Eulerian Gaussian beam method. We do a forward ray tracing using

3

(1.4)-(1.5) to determine the centers of the beams. Level set functions will only be evaluated around these locations, thus allowing a *local* level set implementation of the Eulerian method, which is significantly less expensive than our original Eulerian method in the phase space [8]. To improve the accuracy of the Gaussian beam summation, we do a backward ray tracing by solving (1.4)-(1.5) backward in time. The overall cost is slightly bigger than a fully Lagrangian method, but the numerical resolution is much better since it avoids the problem of ray diverging in a full or semi-Lagrangian setting.

We then generalize and study this semi-Eulerian Gaussian beam method to a higher (third) order Gaussian beam method. The higher order Lagrangian Gaussian beam methods were proposed in [18, 21]. The order beyond Hessian in the Taylor expansion of the phase T can also be found by derivatives of the complex-valued level set functions in the Eulerian framework [13]. We give the details of a third order Gaussian beam method for the linear Schrodinger equation in both Lagrangian and Eulerian frameworks, and implement the semi-Eulerian method for its efficient computation.

The paper is organized as follows. In Section 2, we introduce the semi-Eulerian Gaussian beam formulation and summarize its properties. The higher order semi-Eulerian Gaussian beam methods are derived and studied in Section 3. Numerical examples are given in Section 4 to test the accuracy and efficiency of the semi-Eulerian Gaussian beam methods. We make some conclusive remarks in Section 5 .

2   The Semi-Eulerian Gaussian beam method

In this section we first describe the semi-Eulerian Gaussian beam algorithm in details, then we discuss the computation of the amplitude. Finally some advantages of this new method are pointed out.

2.1   The semi-Eulerian Gaussian beam algorithm

- Step 1. Solve the *forward* ray tracing equations to obtain the positions of centers of all the discrete beams at time t,

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = \boldsymbol{p}, \tag{2.1}$$

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\nabla_{\boldsymbol{y}}V, \tag{2.2}$$

with the initial conditions

$$\boldsymbol{y}(0, \boldsymbol{y}_0^j) = \boldsymbol{y}_0^j,$$
$$\boldsymbol{p}(0, \boldsymbol{y}_0^j) = \nabla_{\boldsymbol{x}}S_0(\boldsymbol{y}_0^j),$$

where $\boldsymbol{y}_0^j$, $j = 1, \cdots, N_{\boldsymbol{y}_0}$ are the equidistant mesh points with mesh size Δy_0, and $N_{\boldsymbol{y}_0}$ is the number of the beams initially centered at $\boldsymbol{y}_0^j$.

4

For each j, we denote $\boldsymbol{p}_0^j = \nabla_{\boldsymbol{x}} S_0(\boldsymbol{y}_0^j)$ and $(\boldsymbol{y}^j, \boldsymbol{p}^j) = \big(\boldsymbol{y}(t, \boldsymbol{y}_0^j), \boldsymbol{p}(t, \boldsymbol{y}_0^j)\big)$, then

$$\phi(t, \boldsymbol{y}^j, \boldsymbol{p}^j) = \phi_0\big(\boldsymbol{y}_0^j, \boldsymbol{p}_0^j\big).$$

The phase function $S(t, \boldsymbol{y}^j, \boldsymbol{p}^j)$ is obtained by solving the ODE

$$\frac{\mathrm{d}S}{\mathrm{d}t} = \frac{1}{2}|\boldsymbol{p}|^2 - V, \tag{2.3}$$

with the initial condition

$$S(0, \boldsymbol{y}_0^j, \boldsymbol{p}_0^j) = S_0(\boldsymbol{y}_0^j).$$

- Step 2. Choose several points $(\boldsymbol{y}^{j_k}, \boldsymbol{p}^{j_k})$, $k = 1, \cdots, K$ around $(\boldsymbol{y}^j, \boldsymbol{p}^j)$ in the $\boldsymbol{y} - \boldsymbol{p}$ phase plane, and solve the *backward* ray tracing equations

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = -\boldsymbol{p}, \tag{2.4}$$

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = \nabla_{\boldsymbol{y}} V, \tag{2.5}$$

with the initial conditions

$$\begin{aligned} \boldsymbol{y}(0, \boldsymbol{y}_0^j) &= \boldsymbol{y}^{j_k}, \\ \boldsymbol{p}(0, \boldsymbol{y}_0^j) &= \boldsymbol{p}^{j_k}. \end{aligned}$$

Denote $(\boldsymbol{y}_0^{j_k}, \boldsymbol{p}_0^{j_k}) = \big(\boldsymbol{y}(t, \boldsymbol{y}^{j_k}), \boldsymbol{p}(t, \boldsymbol{y}^{j_k})\big)$, then

$$\phi(t, \boldsymbol{y}^{j_k}, \boldsymbol{p}^{j_k}) = \phi_0(\boldsymbol{y}_0^{j_k}, \boldsymbol{p}_0^{j_k}). \tag{2.6}$$

- Step 3. Compute $\nabla_{\boldsymbol{y}} \phi$ and $\nabla_{\boldsymbol{p}} \phi$ at $(\boldsymbol{y}^j, \boldsymbol{p}^j)$ by applying some finite difference scheme on $\phi(t, \boldsymbol{y}^j, \boldsymbol{p}^j)$ and $\phi(t, \boldsymbol{y}^{j_k}, \boldsymbol{p}^{j_k})$, where $j = 1, \cdots, N_{\boldsymbol{y}_0}$, $k = 1, \cdots, K$.

- Step 4. Compute the Hessian M by (1.10) and A at $(\boldsymbol{y}^j, \boldsymbol{p}^j)$ by

$$\frac{\mathrm{d}A}{\mathrm{d}t} = -\frac{1}{2}\mathrm{tr}(M)A, \tag{2.7}$$

with the initial condition

$$A(0, \boldsymbol{y}^j, \boldsymbol{p}^j) = A_0(\boldsymbol{y}_0^j).$$

Since solving (2.7) also needs the information of the Hessian M at previous time steps, we give several detailed options for computing A below in Section 2.2.

- Step 5. The approximate Gaussian beam solution is given by the following discrete summation formula,

$$\Phi_{se}^{\varepsilon}(t, \boldsymbol{x}) = \sum_{j=1}^{N_{\boldsymbol{y}_0}} \left(\frac{1}{2\pi\varepsilon} \right)^{\frac{n}{2}} r_{\theta}(\boldsymbol{x} - \boldsymbol{y}^j) \varphi_{se}^{\varepsilon}(t, \boldsymbol{x}, \boldsymbol{y}^j, \boldsymbol{p}^j) \Delta \boldsymbol{y}_0, \qquad (2.8)$$

with $r_{\theta} \in C_0^{\infty}(\mathbb{R}^n)$, $r_{\theta} \geq 0$, a truncation function with $r_{\theta} \equiv 1$ in a ball of radius $\theta > 0$ about the origin and

$$\varphi_{se}^{\varepsilon}(t, \boldsymbol{x}, \boldsymbol{y}^j, \boldsymbol{p}^j) = A(t, \boldsymbol{y}^j, \boldsymbol{p}^j) e^{iT(t, \boldsymbol{x}, \boldsymbol{y}^j, \boldsymbol{p}^j)/\varepsilon}, \qquad (2.9)$$

where

$$\begin{aligned} T(t, \boldsymbol{x}, \boldsymbol{y}^j, \boldsymbol{p}^j) = \quad & S(t, \boldsymbol{y}^j, \boldsymbol{p}^j) + \boldsymbol{p}^j \cdot (\boldsymbol{x} - \boldsymbol{y}^j) \\ & + \frac{1}{2} (\boldsymbol{x} - \boldsymbol{y}^j)^{\top} M(t, \boldsymbol{y}^j, \boldsymbol{p}^j)(\boldsymbol{x} - \boldsymbol{y}^j). \end{aligned}$$

2.2 Computing the amplitude A

- Method 1. Solving (2.7) with some ODE solver.

Remark 2.1 *Because T is divided by ε in (2.9) and each $\boldsymbol{x} - \boldsymbol{y}$ contributes $O(\sqrt{\varepsilon})$, the time step for step 2 and step 4 (computing M and A) is larger than for step 1. For example, if we use a fourth order scheme and require the accuracy to be $O(\varepsilon^2)$, then the time step for (2.4)-(2.5) and (2.7) is $O(\sqrt{\varepsilon})$, while the time step for (2.1)-(2.3) is $O(\varepsilon^{3/4})$. Note that the computation of M only needs time step to be $O(\sqrt{\varepsilon})$ since it is being multiplied by terms of order $|\boldsymbol{x} - \boldsymbol{y}|^2 = O(\varepsilon)$. In general, if we use an s_1-th order numerical scheme and require the accuracy to be $O(\varepsilon^{s_2})$, the time step for (2.4)-(2.5) and (2.7) is $O(\varepsilon^{\frac{s_2}{s_1}})$ and the time step for (2.1)-(2.3) is $O(\varepsilon^{\frac{s_2+1}{s_1}})$.*

Denote Δt_R as the time step for solving (2.4)-(2.5) and (2.7) and $t^{\ell} = \ell \Delta t_R$, $\ell = 1, \cdots, L$, then to solve (2.4)-(2.5) and (2.7) numerically one only needs the information of the Hessian M for each t^{ℓ} which could be computed by one of the following two options:

- Option A. First solve (2.1)-(2.2) starting at $(\boldsymbol{y}_0^j, \boldsymbol{p}_0^j)$ up to the time t^L and record $(\boldsymbol{y}^{j,\ell}, \boldsymbol{p}^{j,\ell})$ at each t^{ℓ}, $\ell = 1, \cdots, L$, then we choose the points $(\boldsymbol{y}^{j_k,\ell}, \boldsymbol{p}^{j_k,\ell})$, $k = 1, \cdots, K$ around $(\boldsymbol{y}^{j,\ell}, \boldsymbol{p}^{j,\ell})$. Next compute $(\boldsymbol{y}_0^{j_k,\ell}, \boldsymbol{p}_0^{j_k,\ell})$ by solving (2.4)-(2.5) with the initial point $(\boldsymbol{y}^{j_k,\ell}, \boldsymbol{p}^{j_k,\ell})$ at time t^{ℓ} and get $\phi(t^{\ell}, \boldsymbol{y}^{j_k,\ell}, \boldsymbol{p}^{j_k,\ell})$ by (2.6), i.e. $\phi(t^{\ell}, \boldsymbol{y}^{j_k,\ell}, \boldsymbol{p}^{j_k,\ell}) = \phi_0(\boldsymbol{y}_0^{j_k,\ell}, \boldsymbol{p}_0^{j_k,\ell})$. Finally the Hessian $M(t^{\ell}, \boldsymbol{y}^{j,\ell}, \boldsymbol{p}^{j,\ell})$ is computed by (1.10) using some difference scheme on $\phi(t^{\ell}, \boldsymbol{y}^{j_k,\ell}, \boldsymbol{p}^{j_k,\ell})$, $k = 1, \cdots, K$.

6

Note that since for each $(\boldsymbol{y}^{j_k,\ell}, \boldsymbol{p}^{j_k,\ell})$, solving (2.4)-(2.5) is independent of each other, one could make use of the parallel algorithms to reduce the computing time.

– Option B. The difference between this option and Option A lies in that, we compute the Hessian adaptively. That is, solve (2.4)-(2.5) with the initial point $(\boldsymbol{y}^{j_k,\ell+1}, \boldsymbol{p}^{j_k,\ell+1})$ for a time interval Δt_R and denote the solution as $(\tilde{\boldsymbol{y}}^{j_k,\ell}, \tilde{\boldsymbol{p}}^{j_k,\ell})$. Then we obtain $\phi(t^\ell, \tilde{\boldsymbol{y}}^{j_k,\ell}, \tilde{\boldsymbol{p}}^{j_k,\ell})$ from $\phi(t^\ell, \boldsymbol{y}^{j_k,\ell}, \boldsymbol{p}^{j_k,\ell})$ by numerical interpolations, which gives $\phi(t^{\ell+1}, \boldsymbol{y}^{j_k,\ell+1}, \boldsymbol{p}^{j_k,\ell+1}) = \phi(t^\ell, \tilde{\boldsymbol{y}}^{j_k,\ell}, \tilde{\boldsymbol{p}}^{j_k,\ell})$ by (2.6).

Remark 2.2 *Option A is easy to implement but more computationally expensive than Option B, while in Option B how to choose the points $(\boldsymbol{y}^{j_k,\ell}, \boldsymbol{p}^{j_k,\ell})$, $k = 1, \cdots, K$ and do the adaptive interpolation efficiently is technically complicated.*

- Method 2. Numerical integration using the Gauss quadrature points on time. Equation (2.7) can be written as

$$
\begin{aligned}
& A(t, \boldsymbol{y}^j, \boldsymbol{p}^j) \\
= \ & A_0(\boldsymbol{y}_0^j, \boldsymbol{p}_0^j) \exp\left(-\frac{1}{2} \int_0^t \mathrm{Tr}M(s, \boldsymbol{y}^{j,s}, \boldsymbol{p}^{j,s}) \mathrm{d}s\right) \\
= \ & A_0(\boldsymbol{y}_0^j, \boldsymbol{p}_0^j) \exp\left(-\frac{1}{2} \sum_{m_1=1}^{M_1} \int_{t^{m_1-1}}^{t^{m_1}} \mathrm{Tr}M(s, \boldsymbol{y}^{j,s}, \boldsymbol{p}^{j,s}) \mathrm{d}s\right) \\
\approx \ & A_0(\boldsymbol{y}_0^j, \boldsymbol{p}_0^j) \exp\left(-\frac{1}{2} \sum_{m_1=1}^{M_1} \left(\sum_{m_2=1}^{M_2} \omega_{m_1,m_2} \mathrm{Tr}\left(M(s, \boldsymbol{y}^{j,t^{m_1,m_2}}, \boldsymbol{p}^{j,t^{m_1,m_2}})\right)\right)\right).
\end{aligned}
$$

Here $t^{m_1} = \frac{m_1 t}{M_1}$, $m_1 = 0, \cdots, M_1$ make an equidistant partition $[t^{m_1-1}, t^{m_1}]$ of time interval $[0, t]$. t^{m_1,m_2} are Gauss quadrature points in the interval $[t^{m_1-1}, t^{m_1}]$, and the coefficients ω_{m_1,m_2} are the corresponding Gauss quadrature weights. The Hessian $M(t^\ell, \boldsymbol{y}^{j,t^{m_1,m_2}}, \boldsymbol{p}^{j,t^{m_1,m_2}})$ can be computed by the method described in Option A of Method 1.

- Method 3. Conservation of the density. The amplitude A can also be computed by

$$A = (\det(\nabla_{\boldsymbol{p}}\phi)^{-1} f)^{1/2}, \tag{2.10}$$

where f satisfies

$$\frac{\mathrm{d}f}{\mathrm{d}t} = 0, \tag{2.11}$$

with the initial condition

$$f_0(\boldsymbol{y}, \boldsymbol{p}) = A_0^2(\boldsymbol{y}, \boldsymbol{p}).$$

(2.11) may be solved in the same way as how we get ϕ in step 1 but with a larger time step (like the one for getting M and A).

However, since ϕ is complexified, the amplitude A in (2.10) is a multi-valued function with single-valued branches. To determine which branch A lies in, we define $\det(\nabla_{\boldsymbol{p}}\phi) = re^{i\eta}, r \in \mathbb{R}^+, \eta \in \mathbb{R}$, and rewrite (2.10) as

$$A = A_0 e^{-\frac{1}{2}(\ln r + i\eta)} = A_0 e^{-\frac{\ln r}{2} - \frac{i\eta}{2}}.$$

By cutting the complex plane along the negative real axis, one knows that the two branches of A are determined by

$$\eta \in (-\pi + 4m\pi, \pi + 4m\pi), \quad \text{and} \quad \eta \in (\pi + 4m\pi, 3\pi + 4m\pi), \quad m \in \mathbb{Z},$$

which correspond to

$$\arg A \in (-\frac{\pi}{2}, \frac{\pi}{2}), \quad \text{and} \quad \arg A \in (\frac{\pi}{2}, \frac{3\pi}{2}),$$

respectively.

On the complex plane, initially $(r, \eta) = (1, 0)$ according to (1.9), then every time it crosses the negative real axis, A gains a phase shift of $\exp(i\pi)$. Therefore as used in [8], one may take the branch $\arg A \in (-\frac{\pi}{2}, \frac{\pi}{2})$ in (2.10) to compute A for a *short* time (i.e. before the first time (r, η) crosses the negative real axis). However, when the evolution time is *long*, the trajectory of (r, η) could become very complicated that one may not be able to decide which branch of A should be taken in (2.10).

Remark 2.3 *Method 2 is more computationally efficient that Method 1 due to the higher order accuracy and less points required for the Gauss quadrature for evaluating the integral. However, Method 1 can be more easily generalized to the higher order semi-Eulerian Gaussian beam method as we will see in the next section. Method 3 works better than the other two for short time, but becomes more complicated in long time.*

2.3 The advantages of the semi-Eulerian Gaussian beam method

1. It inherits from the Eulerian Gaussian beam method that the Hessian is computed by the derivatives of the level set functions (1.10) instead of the Riccati equation (or the dynamic ray tracing equations). This greatly reduces the number of equations to be solved.

2. The choice of the points $(\boldsymbol{y}^{j_k}, \boldsymbol{p}^{j_k})$, $k = 1, \cdots, K$ for computing $\nabla_{\boldsymbol{y}}\phi$ and $\nabla_{\boldsymbol{p}}\phi$ is totally flexible, depending on the numerical difference scheme to be used and the accuracy requirement. For example, the points $(\boldsymbol{y}^{j_k}, \boldsymbol{p}^{j_k})$ can be picked equidistantly to $(\boldsymbol{y}^j, \boldsymbol{p}^j)$ so that the

central difference scheme can be applied. Moreover, this allows computing the level set functions locally by (2.4)-(2.6), thus providing a *local level set implementation* of the Eulerian Gaussian beam method of [8].

3. Moreover, since the computation of $\boldsymbol{\phi}(t, \boldsymbol{y}^j, \boldsymbol{p}^j)$, $j = 1, \cdots, N_{\boldsymbol{y}_0}$ is independent of each other, one may use parallel computations. In other words, this semi-Eulerian Gaussian beam method increases the algorithm's parallellizability compared to the traditional Lagrangian method, because when solving the Riccati equation or the dynamic ray tracing system, it couples all the n^2 components of the Hessian so that each of the components can not be parallel solved.

4. Since Δy_0 is uniform, thus this method offers a better numerical resolution around caustics than the semi-Lagrangian method of [12], and avoids the complicated treatment near the caustics of the method in [8] in the Gaussian beam summation (1.6).

3   Higher order Gaussian beam formulation

In this section we turn to the higher order semi-Eulerian Gaussian beam methods. To give a better illustration we only describe the idea in the one dimensional case up to the third order accuracy, while the derivation of the formulation with a higher order accuracy and in higher dimension is essentially similar but intricately involve with higher order derivatives and higher dimensional tensors. Since so far there is no detailed derivation of the higher order Eulerian Gaussian beam formulation, we arrange this section by the following sequence:

- First, we briefly review the third order Lagrangian formulation for the Schrödinger equation as presented in [21]. By introducing some new quantities, we rewrite the equations for the higher order Hessians which will be helpful for the derivation of the Eulerian formulation. The stability properties of the amplitude equations which will be used in the semi-Eulerian formulation are discussed.

- Then we derive the third order Eulerian formulation systematically.

- Finally we introduce the third order semi-Eulerian algorithm.

3.1   Lagrangian formulation

Write the third order Lagrangian Gaussian beam ansatz as

$$\varphi_{la}(t, x, y_0) = A(t, x, y)e^{iT(t,x,y)/\varepsilon}, \tag{3.1}$$

9

where $y = y(t, y_0)$, A and T are given by the following expressions

$$
\begin{aligned}
A(t, x, y) &= A_0(t, x, y) + \frac{\varepsilon}{i} A_1(t, x, y) \\
&= \left(A_{00}(t, y) + (x - y)A_{01}(t, y) + \frac{1}{2}(x - y)^2 A_{02}(t, y) \right) + \frac{\varepsilon}{i} A_{10}(t, y),
\end{aligned}
\tag{3.2}
$$

$$
\begin{aligned}
T(t, x, y) &= T_0(t, y) + (x - y)T_1(t, y) + \frac{1}{2}(x - y)^2 T_2(t, y) + \frac{1}{6}(x - y)^3 T_3(t, y) + \\
&\quad \frac{1}{24}(x - y)^4 T_4(t, y).
\end{aligned}
\tag{3.3}
$$

Then one has (cf. [18, 21])

$$
\frac{\mathrm{d}y}{\mathrm{d}t} = T_1(t, y), \tag{3.4}
$$

$$
\frac{\mathrm{d}T_0}{\mathrm{d}t} = \frac{1}{2}T_1^2 - V, \tag{3.5}
$$

$$
\frac{\mathrm{d}T_1}{\mathrm{d}t} = -V_x, \tag{3.6}
$$

$$
\frac{\mathrm{d}T_2}{\mathrm{d}t} = -T_2^2 - V_{xx}, \tag{3.7}
$$

$$
\frac{\mathrm{d}T_3}{\mathrm{d}t} = -3T_2T_3 - V_{xxx}, \tag{3.8}
$$

$$
\frac{\mathrm{d}T_4}{\mathrm{d}t} = -4T_2T_4 - 3T_3^2 - V_{xxxx}, \tag{3.9}
$$

$$
\frac{\mathrm{d}A_{00}}{\mathrm{d}t} = -\frac{1}{2}T_2 A_{00}, \tag{3.10}
$$

$$
\frac{\mathrm{d}A_{01}}{\mathrm{d}t} = -\frac{1}{2}T_3 A_{00} - \frac{3}{2}T_2 A_{01}, \tag{3.11}
$$

$$
\frac{\mathrm{d}A_{02}}{\mathrm{d}t} = -\frac{1}{2}T_4 A_{00} - 2T_3 A_{01} - \frac{5}{2}T_2 A_{02}, \tag{3.12}
$$

$$
\frac{\mathrm{d}A_{10}}{\mathrm{d}t} = -\frac{1}{2}A_{02} - \frac{1}{2}T_2 A_{10}. \tag{3.13}
$$

Now we discuss the stability of the initial value problem to (3.10)-(3.13). This stability is important for the Lagrangian method bus was not studied in [18, 21].

Theorem 3.1 *The initial value problem to equations (3.10)-(3.13) are stable for all $t > 0$.*

Proof: Since (3.10)-(3.13) can be solved one after another, the off-diagonal part of the right hand side serves as a forcing term, thus do not affect the stability of this system. Thus for stability of the initial value problem of (3.10)-(3.13) it suffices to drop the off-diagonal terms, and study the

10

following system

$$\begin{aligned}
\frac{\mathrm{d}A_{00}}{\mathrm{d}t} &= -\frac{1}{2}T_2 A_{00}, \\
\frac{\mathrm{d}\tilde{A}_{01}}{\mathrm{d}t} &= -\frac{3}{2}T_2 \tilde{A}_{01}, \\
\frac{\mathrm{d}\tilde{A}_{02}}{\mathrm{d}t} &= -\frac{5}{2}T_2 \tilde{A}_{02}, \\
\frac{\mathrm{d}\tilde{A}_{10}}{\mathrm{d}t} &= -\frac{1}{2}T_2 \tilde{A}_{10}.
\end{aligned}$$

This system can be integrated analytically to get

$$\begin{aligned}
A_{00}(t) &= A_{00}(0)e^{-\frac{1}{2}\int_0^t T_2(s)\mathrm{d}s}, \\
\tilde{A}_{01}(t) &= \tilde{A}_{01}(0)e^{-\frac{3}{2}\int_0^t T_2(s)\mathrm{d}s}, \\
\tilde{A}_{02}(t) &= \tilde{A}_{02}(0)e^{-\frac{5}{2}\int_0^t T_2(s)\mathrm{d}s}, \\
\tilde{A}_{10}(t) &= \tilde{A}_{10}(0)e^{-\frac{1}{2}\int_0^t T_2(s)\mathrm{d}s}.
\end{aligned}$$

Although the sign of T_2 is indefinite, as we showed in the Theorem 2.1 of [8], one can see that

$$\left| e^{-\frac{1}{2}\int_0^t T_2(s)\mathrm{d}s} \right| = |W_{22}|^{-\frac{1}{2}} < C, \quad \text{for all } t > 0,$$

where C is a positive constant. Therefore $A_{00}, \tilde{A}_{01}, \tilde{A}_{02}$ and $\tilde{A}_{10}$ are all bounded for $t > 0$ which implies the stabilities of (3.10)-(3.13). $\square$

One idea of the classical Gaussian beam method is to replace the nonlinear Riccati equation by a system of linear dynamic ray tracing equations using suitable change of variables. Analogously we introduce the following

11

linear system which is equivalent to the Riccati type systems (3.7)-(3.9).

$$\frac{\mathrm{d}W_{21}}{\mathrm{d}t} = V_{xx}W_{22}, \tag{3.14}$$

$$\frac{\mathrm{d}W_{22}}{\mathrm{d}t} = -W_{21}, \tag{3.15}$$

$$\frac{\mathrm{d}W_{31}}{\mathrm{d}t} = 2V_{xx}W_{32} + V_{xxx}W_{22}, \tag{3.16}$$

$$\frac{\mathrm{d}W_{32}}{\mathrm{d}t} = -W_{31} + V_{xx}W_{33}, \tag{3.17}$$

$$\frac{\mathrm{d}W_{33}}{\mathrm{d}t} = -2W_{32}, \tag{3.18}$$

$$\frac{\mathrm{d}W_{41}}{\mathrm{d}t} = 3V_{xx}W_{42} + 3V_{xxx}W_{32} + V_{xxxx}W_{22}, \tag{3.19}$$

$$\frac{\mathrm{d}W_{42}}{\mathrm{d}t} = -W_{41} + 2V_{xx}W_{43} + V_{xxx}W_{33}, \tag{3.20}$$

$$\frac{\mathrm{d}W_{43}}{\mathrm{d}t} = -2W_{42} + V_{xx}W_{44}, \tag{3.21}$$

$$\frac{\mathrm{d}W_{44}}{\mathrm{d}t} = -3W_{43}. \tag{3.22}$$

By some tedious calculation one knows that the following quantities satisfy (3.7)-(3.9) exactly,

$$T_2 = -\frac{W_{21}}{W_{22}}, \tag{3.23}$$

$$T_3 = -\frac{W_{31} + 2T_2W_{32} + T_2^2W_{33}}{W_{22}}, \tag{3.24}$$

$$T_4 = -\frac{W_{41} + 3T_3W_{32} + 3T_2W_{42} + 3T_2^2W_{43} + 3T_2T_3W_{33} + T_2^3W_{44}}{W_{22}}. \tag{3.25}$$

3.2 An Eulerian formulation

Taking the derivatives of (1.8) with x and p up to the third order yields

$$
\begin{aligned}
\mathcal{L}\phi_x &= V_{xx}\phi_p, \\
\mathcal{L}\phi_p &= -\phi_x, \\
\mathcal{L}\phi_{xx} &= 2V_{xx}\phi_{xp} + V_{xxx}\phi_p, \\
\mathcal{L}\phi_{xp} &= -\phi_{xx} + V_{xx}\phi_{pp}, \\
\mathcal{L}\phi_{pp} &= -2\phi_{xp}, \\
\mathcal{L}\phi_{xxx} &= 3V_{xx}\phi_{xxp} + 3V_{xxx}\phi_{xp} + V_{xxxx}\phi_p, \\
\mathcal{L}\phi_{xxp} &= -\phi_{xxx} + 2V_{xx}\phi_{xpp} + V_{xxx}\phi_{pp}, \\
\mathcal{L}\phi_{xpp} &= -2\phi_{xxp} + V_{xx}\phi_{ppp}, \\
\mathcal{L}\phi_{ppp} &= -3\phi_{xpp},
\end{aligned}
$$

12

which implies, according to (3.14)-(3.22),

$$W_{21} = \phi_x, \quad W_{22} = \phi_p, \tag{3.26}$$

$$W_{31} = \phi_{xx}, \quad W_{32} = \phi_{xp}, \quad W_{33} = \phi_{pp}, \tag{3.27}$$

$$W_{41} = \phi_{xxx}, \quad W_{42} = \phi_{xxp}, \quad W_{43} = \phi_{xpp}, \quad W_{44} = \phi_{ppp}. \tag{3.28}$$

Remark 3.2 *Some intuition for constructing the Hessians T_r using W_{rs} $(2 \le l \le 4, \ 1 \le m \le l)$ could be found by differentiating the level set function with respect to x up to the third order,*

$$\phi(t, x, p) = 0, \quad at \quad p = u(x) = \partial_x S(x), \tag{3.29}$$

which gives

$$\phi_x + S_{xx}\phi_p = 0,$$
$$\phi_{xx} + 2S_{xx}\phi_{xp} + S_{xxx}\phi_p + S_{xx}^2\phi_{pp} = 0,$$
$$\phi_{xxx} + 3S_{xxx}\phi_{xp} + 3S_{xx}\phi_{xxp} + 3S_{xx}^2\phi_{xpp} + S_{xxxx}\phi_p + 3S_{xx}S_{xxx}\phi_{pp} + S_{xx}^3\phi_{ppp} = 0.$$

One may notice that T_l behaves the same as the l-th order x-derivative of the phase function S. Therefore it is easy to represent the even higher order Hessians T_l, $l \ge 5$ using some new W_{lm}, $1 \le m \le l$ by differentiating (3.29) more times, although the justification of the equations for W_{lm} is going to involve more tedious calculations, and is omitted here.

Equations (3.5) and (3.10)-(3.13) imply that the Eulerian formulation for the phase and amplitudes are

$$\mathcal{L}T_0 = \frac{1}{2}T_1^2 - V,$$

$$\mathcal{L}A_{00} = -\frac{1}{2}T_2 A_{00},$$

$$\mathcal{L}A_{01} = -\frac{1}{2}T_3 A_{00} - \frac{3}{2}T_2 A_{01},$$

$$\mathcal{L}A_{02} = -\frac{1}{2}T_4 A_{00} - 2T_3 A_{01} - \frac{5}{2}T_2 A_{02},$$

$$\mathcal{L}A_{10} = -\frac{1}{2}A_{02} - \frac{1}{2}T_2 A_{10},$$

where T_l, $l = 2, 3, 4$ are given by (3.23)-(3.25) and (3.26)-(3.28).

3.3 Semi-Eulerian formulation

The first two steps of the third order semi-Eulerian Gaussian beam method are essentially the same as that described in Section 2, so we only update Step 3, 4 and 5 here.

13

- Step 3. Compute W_{lm}, $2 \leq l \leq 4$, $1 \leq m \leq l$ using (3.26)-(3.28) by applying finite difference schemes on $\phi(t, \boldsymbol{y}^j, \boldsymbol{p}^j)$ and $\phi(t, \boldsymbol{y}^{j_k}, \boldsymbol{p}^{j_k})$, where $j = 1, \cdots, N_{\boldsymbol{y}_0}$, $k = 1, \cdots, K$.

- Step 4. Computing the Hessians T_l, $l = 2, 3, 4$ by (3.23)-(3.25) and the amplitudes by (3.10)-(3.13) using Method 1 introduced in Section 2.2 at points (y^j, p^j).

- Step 5. The approximate Gaussian beam solution is given by the following discrete summation formula,

$$\Phi_{se}^\varepsilon(t, x) = \sum_{j=1}^{N_{y_0}} \left(\frac{1}{2\pi\varepsilon} \right)^{\frac{1}{2}} r_\theta(x - y^j) \varphi_{se}^\varepsilon(t, x, y^j, p^j) \Delta y_0, \qquad (3.30)$$

where Δy_0 is the mesh size for the uniformly distributed points y_0^j, $r_\theta \in C_0^\infty(\mathbb{R})$, $r_\theta \geq 0$ is a truncation function with $r_\theta \equiv 1$ in a ball of radius $\theta > 0$ about the origin and

$$\varphi_{se}^\varepsilon(t, x, y^j, p^j) = A(t, x, y^j, p^j) e^{iT(t,x,y^j,p^j)/\varepsilon}, \qquad (3.31)$$

where A and T are given by

$$
\begin{aligned}
A(t, x, y^j, p^j) &= \left(A_{00}(t, y^j, p^j) + (x - y^j) A_{01}(t, y^j, p^j) \right. \\
&\quad \left. + \frac{1}{2}(x - y^j)^2 A_{02}(t, y^j, p^j) \right) + \frac{\varepsilon}{i} A_{10}(t, y^j, p^j), \\
T(t, x, y^j, p^j) &= T_0(t, y^j, p^j) + (x - y^j) T_1(t, y^j, p^j) + \frac{1}{2}(x - y^j)^2 T_2(t, y^j, p^j) \\
&\quad + \frac{1}{6}(x - y^j)^3 T_3(t, y^j, p^j) + \frac{1}{24}(x - y^j)^4 T_4(t, y^j, p^j).
\end{aligned}
$$

Remark 3.3 *The higher order semi-Eulerian Gaussian beam method has the following properties:*

1. *We use (3.23)-(3.25) and (3.26)-(3.28) to compute the T_l, $l = 2, 3, 4$, therefore it removes the stability constrain of solving (3.7)-(3.9) (or (3.14)-(3.22)) directly. Since the real part of T_2 is possibly negative, it usually requires implicit numerical solvers to guarantee the stabilities of (3.7)-(3.9) which greatly increases the algorithm complexity and computing time. The stability constraint becomes even worse when one goes to the high dimensional cases.*

2. *The time step for solving (2.1)-(2.3) needs to be smaller than the one for (2.4)-(2.5) and (3.10)-(3.13) due to the accuracy requirement as we have discussed in Remark 2.1.*

3. *For higher dimension problems, the semi-Eulerian Gaussian beam method can be extended straightforwardly by taking gradient of* (1.8) *with respect to $\boldsymbol{x}$ and $\boldsymbol{p}$. Moreover, the semi-Eulerian Gaussian beam method for higher dimension and higher order cases can be much more effective since the direct computation of nonlinear ODE systems* (3.7)-(3.9) *for $T_l(2 \le l \le 4)$ is rather difficult and the computational cost to a large ODE systems* (3.14)-(3.22) *for W_{lm} is also extremely high.*

Remark 3.4 *As discussed in [21], the Gaussian beam solution* (3.1)-(3.3) *may break down when higher order terms in the phase function T_l $(l \ge 3)$ is considered. For example in 1d, the imaginary part of T_l $(l \ge 3)$ does not remain positive, this makes the solution no longer a Gaussian profile. A simple approach provided in [21] is to expand higher order exponential terms in the form of powers series, e.g. the third order Gaussian beam solution can be rewritten as*

$$\varphi = (A_{00} + \tilde{x}A_{01} + \frac{1}{2}\tilde{x}^2 A_{02} + \frac{\varepsilon}{i}A_{10}) \exp\left(\frac{i}{\varepsilon}(T_0 + \tilde{x}T_1 + \frac{1}{2}\tilde{x}^2 T_2)\right) \times$$
$$\left(1 + \frac{i}{\varepsilon}\frac{T_3}{6}\tilde{x}^3 + \frac{1}{2}(\frac{i}{\varepsilon}\frac{T_3}{6}\tilde{x}^3)^2\right) \times \left(1 + \frac{i}{\varepsilon}\frac{T_4}{24}\tilde{x}^4\right)$$

in which $\tilde{x} = x - y$.

4   Numerical examples

In this section we give three numerical examples to verify the accuracy and efficiency of the semi-Eulerian Gaussian beam method. In the first and third example, as potential $V = 0$, the reference solution Ψ^ε of the Schrödinger equation (1.1) can be computed as

$$\Psi^\varepsilon(t, \boldsymbol{x}) = \mathcal{F}^{-1}\left(\mathcal{F}(\Psi^\varepsilon(0, \boldsymbol{x}))e^{i\frac{\varepsilon|\boldsymbol{k}|^2}{2}t}\right),$$

here $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier and the inverse Fourier transforms, and $\boldsymbol{k}$ is the Fourier variable. This solution could be approximated using the subroutines *fft* and *ifft* in Matlab. In the second example, the reference solution Ψ^ε to Schödinger equation (1.1) is obtained by the Strang splitting spectral method [1] with a very fine mesh and a very small time step. In all examples, we choose a suitably large domain so that the periodic boundary condition does not introduce a significant error to the initial value problem. The truncation parameter θ in (2.8) is chosen fairly large as we discussed in [8].

Example 1 (1d): We take $V = 0$, $A_0 = e^{-25x^2}$, $S_0 = \frac{1}{\pi}\cos(\pi x)$ in (1.2). Then

$$\phi(0, y, p) = -iy + (p - S_0'(y)) = -iy + p + \sin(\pi y),$$
$$\phi(t, y, p) = -i(y - pt) + p + \sin(\pi(y - pt)).$$

| ε	$\frac{1}{256}$	$\frac{1}{512}$	$\frac{1}{1024}$	$\frac{1}{2048}$	$\frac{1}{4096}$
1st order	2.02×10^{-2}	1.08×10^{-2}	5.62×10^{-3}	2.86×10^{-3}	1.44×10^{-3}
2nd order	1.51×10^{-2}	9.03×10^{-3}	4.92×10^{-3}	2.57×10^{-3}	1.31×10^{-3}
3rd order	5.58×10^{-3}	1.81×10^{-3}	5.09×10^{-4}	1.35×10^{-4}	3.48×10^{-5}
1st order	2.17×10^{-2}	1.08×10^{-2}	5.62×10^{-3}	2.86×10^{-3}	1.44×10^{-3}
2nd order	2.06×10^{-2}	1.11×10^{-2}	5.82×10^{-3}	2.97×10^{-3}	1.50×10^{-3}
3rd order	1.26×10^{-2}	3.69×10^{-3}	1.00×10^{-3}	2.60×10^{-4}	6.66×10^{-5}

Table 1: The l^2 errors of the first, second and third order semi-Eulerian Gaussian beam methods for Example 1. The first three lines are about the initial l^2 errors at $t = 0$, and the second three lines are about the l^2 errors at $t = 0.5$. The convergence rate in ε are 0.9784 of 1st order, 0.9449 of 2nd order and 1.8909 of 3rd order.

Thus the quantities $T_l, 2 \leq l \leq 4$, can be solved analytically by using relation (3.23)-(3.28). The amplitude A is computed by Method 1 (Option A) in Section 2.2, by using the fourth order Runge-Kutta method as the ODE solver. The time step Δt in (3.10)-(3.13) and the number of beams N_{y_0} are taken according to $\Delta t \sim \varepsilon^{-1/2}$ and $N_{y_0} \sim \varepsilon^{-1/2}$, e.g. the matched choices of $(\varepsilon, \Delta t, N_{y_0})$ can be $(\frac{1}{1024}, \frac{1}{128}, 128)$, $(\frac{1}{4096}, \frac{1}{256}, 256)$. The output time is $t = 0.5$.

The l^2 numerical errors of the first, second and third order semi-Eulerian Gaussian beam method are given in Table 1. The l^1, l^2, l^∞ errors of the third order method are given in Table 2. We plot the wave amplitudes and the absolute errors in Figure 1 for different ε. One can see that the 1st and 2nd order Gaussian beam solutions give nearly the same accuracy of $O(\varepsilon)$ in l^2 norm, and the 3rd order Gaussian beam solution gives an accuracy of almost $O(\varepsilon^2)$ in l^2 norm. This agrees with the error cancelation phenomenon during the beam summation discussed in [15, 18].

Example 2 (1d): We take $V = \frac{1}{2}x^2$, $A_0 = e^{-10x^2}$, $S_0 = -\frac{1}{2}\log(2\cosh(2x))$ in (1.2) and output the solution at $t = 0.65$. We use the fourth order Runge-Kutta method to solve the ODE system (2.1)-(2.2), (2.4)-(2.5), and the time step Δt here satisfies $\Delta t \sim \varepsilon^{-3/4}$. The quantities T_l $(2 \leq l \leq 4)$ are obtained by using the central difference scheme to evaluate (3.23)-(3.28). We use the same method as in Example 1 to compute the amplitude A. The time step Δt in (3.10)-(3.13) and number of beams N_{y_0} are chosen the same with Example 1.
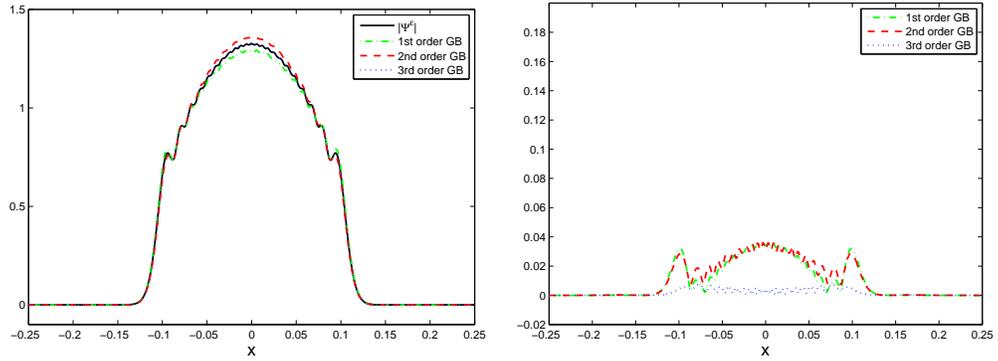
The l^2 numerical errors of the first, second and third order semi-Eulerian Gaussian beam method are given in Table 3. The l^1, l^2, l^∞ errors of the third order method are given in Table 4. We plot the wave amplitudes and the absolute errors in Figure 2 for different ε. The convergence rate is almost
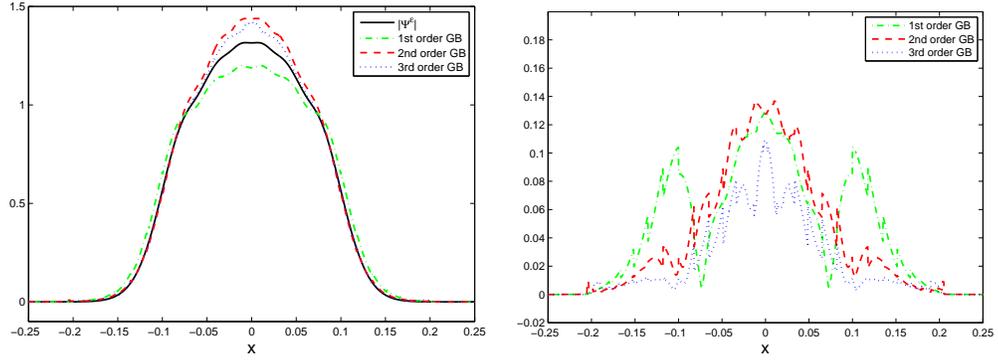
(a) $\varepsilon = \frac{1}{256}$
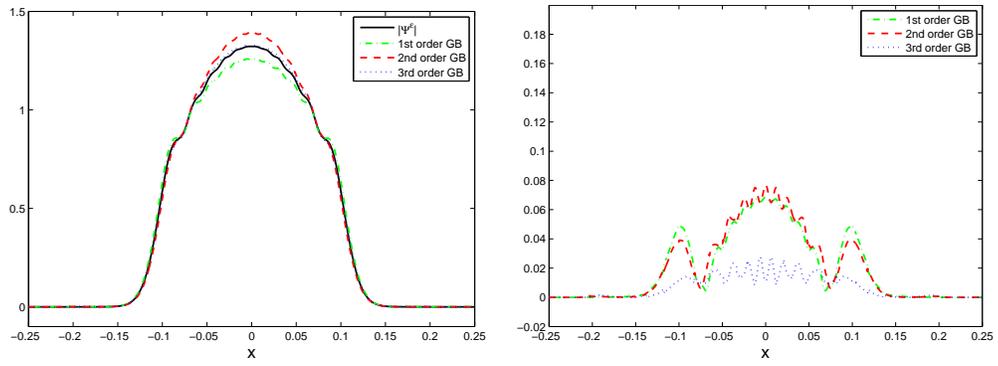
(b) $\varepsilon = \frac{1}{512}$

(c) $\varepsilon = \frac{1}{1024}$

Figure 1: Example 1, the semi-Eulerian Gaussian beam solution in different order versus the 'exact' solution for $\varepsilon = \frac{1}{256}, \frac{1}{512}, \frac{1}{1024}$. The left figures are the comparisons of wave amplitude at $t = 0.5$; the right figures plot the l^2-error.

17

ε	$\frac{1}{256}$	$\frac{1}{512}$	$\frac{1}{1024}$	$\frac{1}{2048}$	$\frac{1}{4096}$
initial l^1 error	2.29×10^{-3}	7.22×10^{-4}	2.00×10^{-4}	5.26×10^{-5}	1.35×10^{-5}
initial l^2 error	5.58×10^{-3}	1.82×10^{-3}	5.09×10^{-4}	1.35×10^{-4}	3.48×10^{-5}
initial l^∞ error	2.45×10^{-2}	8.27×10^{-3}	2.35×10^{-3}	6.27×10^{-4}	1.62×10^{-4}
final l^1 error	2.86×10^{-3}	8.80×10^{-4}	2.24×10^{-4}	5.39×10^{-5}	1.31×10^{-5}
final l^2 error	1.26×10^{-2}	3.69×10^{-3}	1.00×10^{-3}	2.60×10^{-4}	6.65×10^{-5}
final l^∞ error	1.09×10^{-1}	2.76×10^{-2}	7.46×10^{-3}	2.47×10^{-3}	7.35×10^{-4}

Table 2: The l^1, l^2, l^∞ errors of the third order semi-Eulerian Gaussian beam method for Example 1. The convergence rate in ε are 1.9426 in the l^1 norm, 1.8909 in the l^2 norm and 1.8031 in the l^∞ norm.

ε	$\frac{1}{256}$	$\frac{1}{512}$	$\frac{1}{1024}$	$\frac{1}{2048}$	$\frac{1}{4096}$
1st order	1.14×10^{-2}	5.94×10^{-3}	3.03×10^{-3}	1.53×10^{-3}	7.71×10^{-4}
2nd order	9.03×10^{-3}	5.03×10^{-3}	2.61×10^{-3}	1.33×10^{-3}	6.70×10^{-4}
3rd order	1.40×10^{-3}	4.20×10^{-4}	1.10×10^{-4}	2.83×10^{-5}	7.15×10^{-6}
1st order	1.14×10^{-2}	5.94×10^{-3}	3.03×10^{-3}	1.53×10^{-3}	7.70×10^{-4}
2nd order	1.02×10^{-2}	5.55×10^{-3}	2.90×10^{-3}	1.48×10^{-3}	7.49×10^{-4}
3rd order	2.73×10^{-3}	8.00×10^{-4}	2.21×10^{-4}	5.86×10^{-5}	1.52×10^{-5}

Table 3: The l^2 errors of the first, second and third order semi-Eulerian Gaussian beam methods for Example 2. The first three lines are about the initial l^2 errors at $t = 0$, and the second three lines are about the final l^2 errors at $t = 0.65$. The convergence rate in ε are 0.9720 of 1st order, 0.9419 of 2nd order and 1.8722 of 3rd order method respectively.

the same as in Example 1.

Example 3 (2d): We take $V = 0$, $A_0 = e^{-25(x_1^2 + x_2^2)}$, $S_0 = \frac{0.45}{\pi}(\sin(\pi x_1) - 1)(\sin(\pi x_2) - 1)$ in (1.2). In this example we study a two dimensional example which presents a pipe-shape caustics as shown in [7]. We use the first order semi-Eulerian Gaussian beam method and output the solution at $t = 1$. The solution of (1.8) is given by
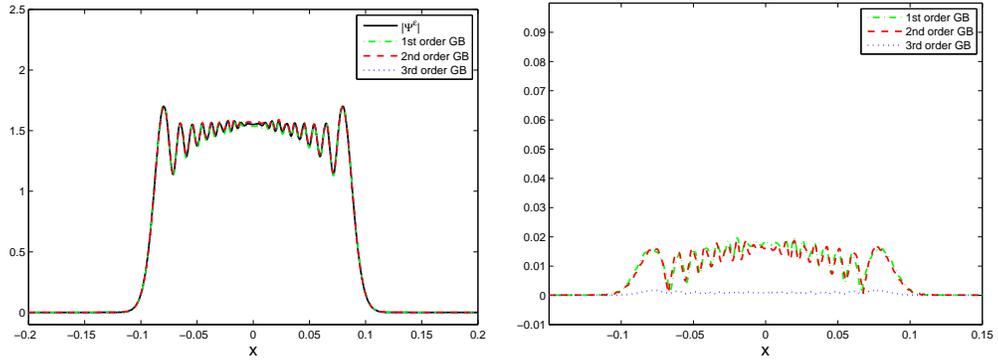
$$
\begin{aligned}
&\phi_1(t, y_1, y_2, p_1, p_2) \\
&= -i(y_1 - p_1 t) + \Big(p_1 - 0.45 \cos(\pi(y_1 - p_1 t))(\sin(\pi(y_2 - p_2 t)) - 1)\Big), \\
&\phi_2(t, y_1, y_2, p_1, p_2) \\
&= -i(y_2 - p_2 t) + \Big(p_2 - 0.45 \cos(\pi(y_2 - p_2 t))(\sin(\pi(y_1 - p_1 t)) - 1)\Big),
\end{aligned}
$$

18
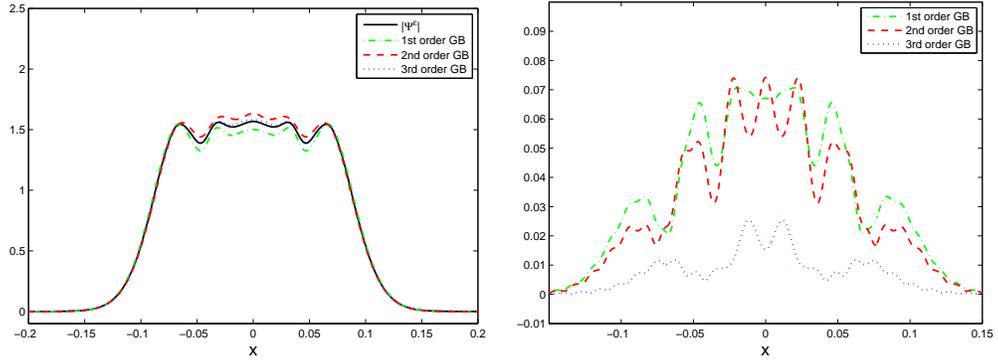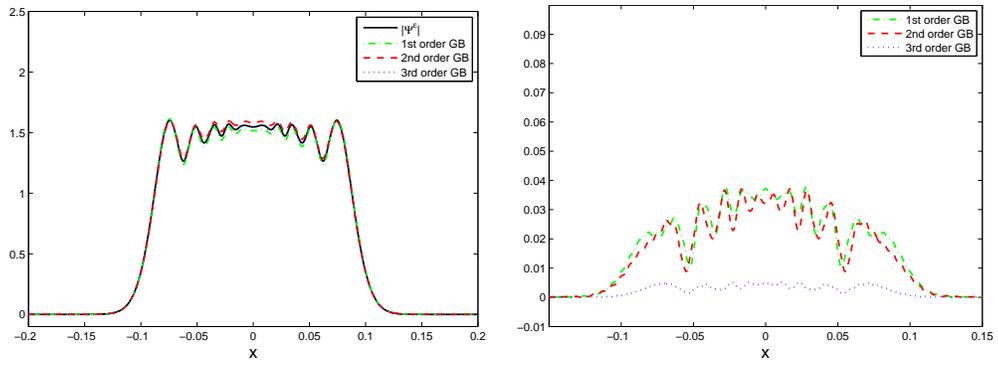
Figure 2: Example 2, the semi-Eulerian Gaussian beam solution in different order versus the 'exact' solution for $\varepsilon = \frac{1}{256}, \frac{1}{512}, \frac{1}{1024}$. The left figures are the comparisons of wave amplitude at $t = 0.65$; the right figures plot the l^2-error.

| ε	$\frac{1}{256}$	$\frac{1}{512}$	$\frac{1}{1024}$	$\frac{1}{2048}$	$\frac{1}{4096}$
initial l^1 error	7.12×10^{-4}	2.07×10^{-4}	5.40×10^{-5}	1.38×10^{-5}	3.49×10^{-6}
initial l^2 error	1.40×10^{-3}	4.20×10^{-4}	1.10×10^{-4}	2.83×10^{-5}	7.15×10^{-6}
initial l^∞ error	4.36×10^{-3}	1.55×10^{-3}	4.10×10^{-4}	1.05×10^{-4}	2.67×10^{-5}
final l^1 error	5.72×10^{-4}	1.74×10^{-4}	4.67×10^{-5}	1.19×10^{-5}	2.98×10^{-6}
final l^2 error	2.73×10^{-3}	8.00×10^{-4}	2.21×10^{-4}	5.86×10^{-5}	1.52×10^{-5}
final l^∞ error	2.53×10^{-2}	5.13×10^{-3}	1.76×10^{-3}	5.76×10^{-4}	1.77×10^{-4}

Table 4: The l^1, l^2, l^∞ errors of the third order semi-Eulerian Gaussian beam method for Example 2. The convergence rate with ε in 1.8961 in the l^1 norm, 1.8722 in the l^2 norm and 1.7898 in the l^∞ norm.

ε	$\frac{1}{256}$	$\frac{1}{512}$	$\frac{1}{1024}$
initial l^1 error	1.23×10^{-2}	6.88×10^{-3}	3.68×10^{-3}
initial l^2 error	3.17×10^{-2}	1.78×10^{-2}	9.53×10^{-3}
initial l^∞ error	1.81×10^{-1}	1.01×10^{-1}	5.37×10^{-2}
final l^1 error	1.83×10^{-2}	7.74×10^{-3}	3.87×10^{-3}
final l^2 error	4.41×10^{-2}	1.98×10^{-2}	9.76×10^{-3}
final l^∞ error	2.67×10^{-1}	1.64×10^{-1}	9.04×10^{-2}

Table 5: The l^1, l^2, l^∞ errors of the first order semi-Eulerian Gaussian beam method for Example 3. The convergence rate with ε are 1.1207 in the l^1 norm, 1.0879 in the l^2 norm and 0.7812 in the l^∞ norm.

which gives T_2 analytically. The amplitude A is obtained by Method 2 in Section 2.2.

The l^1, l^2, l^∞ errors of the first order method at $t = 1$ are given in Table 5. We plot the comparison of the wave amplitudes in Figure 3 for $\varepsilon = \dfrac{1}{512}, \dfrac{1}{1024}$.

5 Conclusion

In this paper, we introduce a semi-Eulerian implementation of the Eulerian Gaussian beam method developed in [8]. By evolving the ray tracing equations, we determine the center of the beams which enables us to compute the level set functions only *locally* thus is much more efficient than the phase-space based Eulerian method. We also trace back in time along the rays so as to improve the numerical accuracy in the Gaussian beam summation. The overall cost is slightly higher than a full Lagrangian method but much
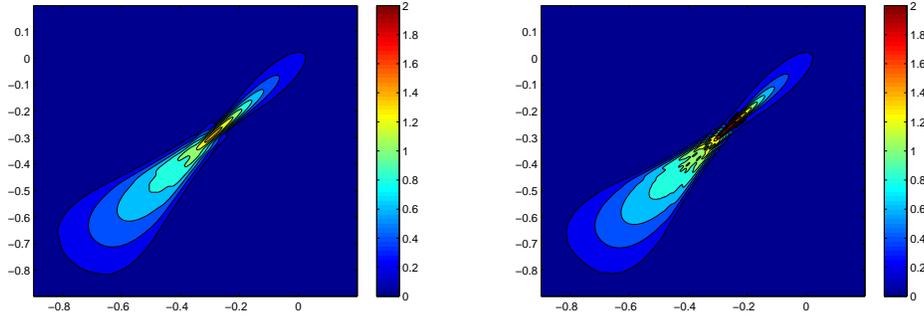
(a) Numerical Solution, Left: $\varepsilon = \frac{1}{512}$, Right: $\varepsilon = \frac{1}{1024}$
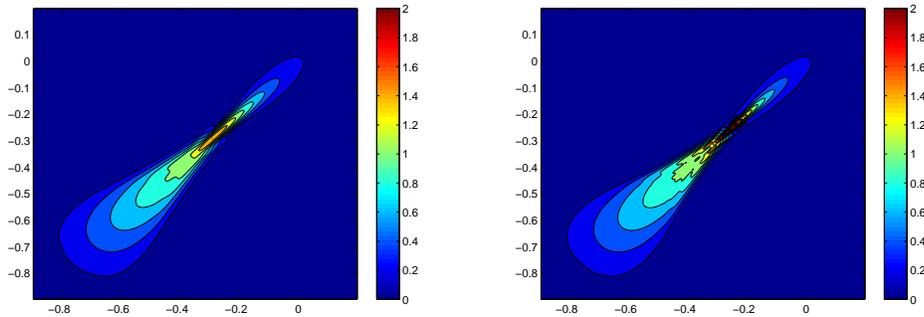
(b) "Exact" Solution, Left: $\varepsilon = \frac{1}{512}$, Right: $\varepsilon = \frac{1}{1024}$
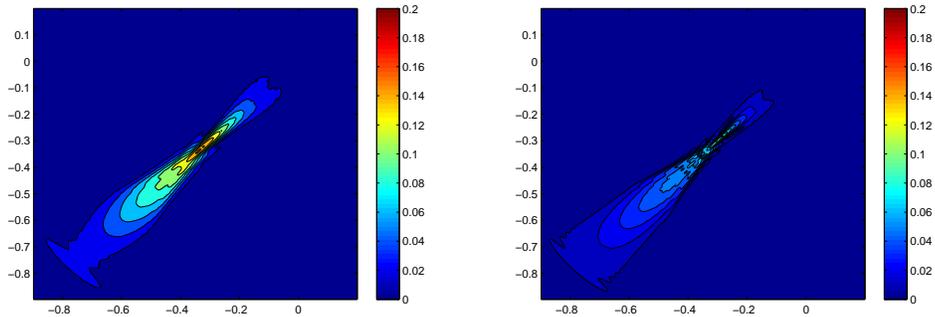
(c) Numerical Error, Left: $\varepsilon = \frac{1}{512}$, Right: $\varepsilon = \frac{1}{1024}$

Figure 3: Example 3, the comparisons of wave amplitudes using the first order semi-Eulerian Gaussian beam method for $\varepsilon = \dfrac{1}{512}, \dfrac{1}{1024}$.

lower than the phase-space Eulerian method. It offers a numerical resolution comparable to the Eulerian method but better than a Lagrangian method since it avoids the problem of diverging rays in a Lagrangian method. We also generalize the semi-Eulerian Gaussian beam method to a third order Gaussian beam method. Several numerical examples are given to show the accuracy and efficiency of the semi-Eulerian and high order Gaussian beam methods.

References

[1] W. Bao, S. Jin and P. A. Markowich, *On time-splitting spectral approximations for the Schrödinger equation in the semiclassical regime*, J. Comp. Phys. 175 (2002), no. 2, 487–524.

[2] 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.

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

[4] E.J. Heller, *Guided Gaussian wave packets*, Accounts of Chemical Research, 39 (2006), no. 2, 127-134.

[5] N.R. Hill, *Gaussian beam migration*, Geophysics, 55 (1990), No. 11, 1416-1428.

[6] N. R. Hill, *Prestack Gaussian-beam depth migration*, Geophysics, 66 (2001), no. 4, 1240-1250.

[7] S. Jin and S. Osher, *A level set method for the computation of multi-valued solutions to quasi-linear hyperbolic PDEs and Hamilton-Jacobi equations*, Comm. Math. Sci., 1 (2003), no. 3, 575-591.

[8] 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), no. 4, 995-1020.

[9] S. Jin, H. Wu, and X. Yang, *A numerical study of the Gaussian beam methods for one-dimensional Schrödinger-Poisson equations*, J. Comp. Math., to appear.

[10] S. Jin, H. Wu, X. Yang and Z.Y. Huang, *Bloch decomposition-based Gaussian beam method for the Schrödinger equation with periodic potentials*, preprint.

[11] S. Leung and J. Qian, *Eulerian Gaussian beams for Schrödinger equations in the semi-classical regime*, J. Comp. Phys., 228 (2009), 2951-2977.

[12] S. Leung, J. Qian and R. Burridge, *Eulerian Gaussian beams for high-frequency wave propagation*, Geophysics, 72 (2007), no. 5, 61-76.

[13] H.L. Liu and J. Ralston, Recovery of high frequency wave fields from phase space based measurements, preprint.

[14] H.L. Liu and J. Ralston, Recovery of high frequency wave fields for the acoustic wave equation, preprint.

[15] M. Motamed and O. Runborg, *Taylor expansion and discretization errors in Gaussian beam superposition*, preprint.

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

[17] J. Ralston, *Gaussian beams and the propagation of singularities*, Studies in PDEs, MAA stud. Math., 23 (1982), 206-248.

[18] N. M. Tanushev, *Superpositions and higher order Gaussian beams*, Comm. Math. Sci., 6 (2008), no. 2, 449-475.

[19] N.M. Tanushev, B. Engquist and R. Tsai, *Gaussian Beam Decomposition of High Frequency Wave Fields*, J. Comp. Phys., to Appear.

[20] N.M.Tanushev, J.Qian, J.V.Ralston, *Mountain Waves and Gaussian Beams*, SIAM MMS, 6 (2007), No.2, 688-709.

[21] D. Yin and C. Zheng, *Fourth order Gaussian beam and interface conditions for the one-dimensional linear Schrödinger equation with singular potentials*, preprint.