Journal of Computational Physics 382 (2019) 27–42

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Geometric exponential integrators

Xuefeng Shen, Melvin Leok *

ARTICLE INFO

Article history:
Received 2 May 2018
Received in revised form 17 November 2018
Accepted 10 January 2019
Available online 21 January 2019

Keywords:
Geometric integrators
Exponential integrators
Symplectic integrators
Hamiltonian systems
Highly oscillatory problems

ABSTRACT

In this paper, we consider exponential integrators for semilinear Poisson systems. Two types of exponential integrators are constructed, one preserves the Poisson structure, and the other preserves energy. Numerical experiments for semilinear Possion systems obtained by semi-discretizing Hamiltonian PDEs are presented. These geometric exponential integrators exhibit better long time stability properties as compared to non-geometric integrators, and are computationally more efficient than traditional symplectic integrators and energy-preserving methods based on the discrete gradient method.

1. Introduction

Hamiltonian partial differential equations are often simulated by semi-discretizing the spatial differential operators, and applying a symplectic or energy-preserving integrator to the resulting system of Hamiltonian ordinary differential equations. The critical challenge associated with such an approach is that the resulting system of differential equations become increasingly stiff as the spatial mesh is refined. This is addressed by combining geometric integrators with exponential integrators [6], which are a class of numerical integrators for stiff systems whose vector field can be decomposed into a linear term and a nonlinear term,

$$\dot{q} = Aq + f(q). \tag{1}$$

Usually, the coefficient matrix A has a large spectral radius, and is responsible for the stiffness of the system of differential equations, while the nonlinear term $f(q)$ is relatively smooth. There are various ways to construct an exponential integrator [10]. For example, we can perform a change of variables $\tilde{q}(t) = e^{-At}q(t)$, and transform (1) to obtain

$$[e^{-At}q(t)]' = \tilde{q}'(t) = e^{-At}f(e^{At}\tilde{q}(t)). \tag{2}$$

Notice that the Jacobian matrix of (2) equals $e^{-At}\nabla f e^{At}$, which has a smaller spectral radius than the Jacobian matrix $A + \nabla f$ of (1). A natural idea is to apply a classical integrator for the mollified system (2) to obtain an approximation of $\tilde{q}(t)$, then invert the change of variables to obtain an approximation of the solution $q(t)$ of (1). In Section 2, we shall demonstrate how to construct symplectic exponential integrators using this approach.

* Corresponding author.
E-mail address: mleok@math.ucsd.edu (M. Leok).

https://doi.org/10.1016/j.jcp.2019.01.005
0021-9991/

Another way of constructing exponential integrators starts from the variation-of-constants formula,

$$q(t) = e^{A(t-t_0)}q(t_0) + \int_{t_0}^{t} e^{A(t-\tau)} f(q(\tau))d\tau, \tag{3}$$

which is the exact solution for (1) with initial condition $q(t_0) = q_0$. Then, a computable approximation can be obtained by approximating the $f(q(\tau))$ term inside the integral. If we approximate $f(q(\tau))$ by $f(q_k)$, we arrive at the exponential Euler method,

$$q_{k+1} = e^{Ah}q_k + \int_0^h e^{A\tau}d\tau \cdot f(q_k). \tag{4}$$

An exponential Runge–Kutta method of collocation type [5] could also be constructed by approximating $f(q(\tau))$ with polynomials. In Section 3, we shall show how to construct energy-preserving exponential integrators from (3).

In this paper, we consider a specific form of (1) which is a Poisson system. We assume $A = JD$, $f(q) = J\nabla V(q)$, where $J^{\mathrm{T}} = -J$, $D^{\mathrm{T}} = D$, and $JD = DJ$, thus the coefficient matrix A is also skew-symmetric. The assumption that J and D commutes turns out to be essential for constructing structure-preserving integrators, and it is naturally satisfied in practical problems. Now, the semilinear system (1) can be written as,

$$\dot{q} = J(Dq + \nabla V(q)) = J\nabla H(q), \tag{5}$$

with Hamiltonian function $H(q) = \frac{1}{2}q^{\mathrm{T}}Dq + V(q)$. Equation (5) describes a constant Poisson system, and there are at least two quantities that are preserved by the flow: the Poisson structure $J_{ij}\frac{\partial}{\partial x_i} \otimes \frac{\partial}{\partial x_j}$ and Hamiltonian $H(q)$. Geometric integrators that preserve the geometric structure and first integrals of the system typically exhibit superior qualitative properties when compared to non-geometric integrators, and they are an active area of research [4,8,9].

In this paper we will introduce the exponential midpoint rule and the energy-preserving exponential integrator. Both of these methods are implicit, and a significant advantage of adopting the exponential integrator approach in the context of geometric integrators is that the resulting geometric exponential integrators can be implemented using fixed point iterations as opposed to the more computationally expensive Newton iterations. Recall that classical implicit Runge–Kutta methods

$$\begin{cases} Y_i = y_n + h \cdot \sum_{j=1}^{s} a_{ij} f(t_n + c_j h, Y_j), \\ y_{n+1} = y_n + h \cdot \sum_{i=1}^{s} b_i f(t_n + c_i h, Y_i), \end{cases}$$

have a form that naturally lends itself to fixed point iterations. However, when $\frac{\partial f}{\partial y}$ has a large spectral radius, the timestep is forced to be very small in order to guarantee that the fixed point iteration converges. The alternative is to use a Newton type iteration, which is time consuming since we need to perform LU decomposition ($O(n^3)$ complexity) during each iteration. This is the problem we face for the stiff semilinear system (1) when the coefficient matrix A has a large spectral radius. In contrast, in both the exponential midpoint rule

$$q_{k+1} = e^{Ah}q_k + h \cdot e^{A\frac{h}{2}} f\Big(\frac{e^{A\frac{h}{2}}q_k + e^{-A\frac{h}{2}}q_{k+1}}{2}\Big),$$

and the energy-preserving exponential integrator

$$q_{k+1} = e^{Ah}q_k + \int_0^h e^{A\tau}d\tau \cdot J\overline{\nabla} V(q_k, q_{k+1}),$$

the matrix A only appears in the exponential term e^{Ah}. Since A is skew-symmetric, this term is an orthogonal matrix, which has spectral radius 1. Thus, fixed point iterations can be used to implement the exponential integrator, regardless of the stiffness of A.

In summary, the main contributions of this paper involve the derivation of geometric exponential integrators that either preserve the Poisson structure or Hamiltonian of (5). They exhibit long time stability, allow for relatively larger timesteps for the stiff problem, and are computationally more efficient as they can be implemented using fixed point iterations as opposed to Newton type iterations. For the rest of the paper, Section 2 is devoted to developing symplectic exponential integrators that preserve the Poisson structure; Section 3 is devoted to developing energy preserving exponential integrators; numerical methods and experiments are presented in Section 4 and Section 5, respectively.

2. Symplectic exponential integrator

For constant Poisson systems (5), it was shown in [15] that the midpoint rule and diagonally implicit symplectic Runge–Kutta methods preserve the Poisson structure $J_{ij}\frac{\partial}{\partial x_i}\otimes\frac{\partial}{\partial x_j}$. We first start by constructing an exponential midpoint rule: apply the classical midpoint rule to the transformed system (2) to obtain

$$\frac{\tilde{q}_{k+1}-\tilde{q}_k}{h}=e^{-At_{k+1/2}}f\left(e^{At_{k+1/2}}\frac{\tilde{q}_{k+1}+\tilde{q}_k}{2}\right), \tag{6}$$

where

$$t_{k+1/2}=\frac{t_k+t_{k+1}}{2},\quad h=t_{k+1}-t_k,\quad \tilde{q}_k=e^{-At_k}q_k,\quad \tilde{q}_{k+1}=e^{-At_{k+1}}q_{k+1}.$$

Transform (6) back to q_k and q_{k+1}, and we obtain the exponential midpoint rule,

$$q_{k+1}=e^{Ah}q_k+h\cdot e^{A\frac{h}{2}}f\left(\frac{e^{A\frac{h}{2}}q_k+e^{-A\frac{h}{2}}q_{k+1}}{2}\right). \tag{7}$$

Theorem 1. *The exponential midpoint rule (7) preserves the Poisson structure when applied to the semilinear Poisson system (5).*

Proof. Recall that a map ϕ preserves Poisson structure $J_{ij}\frac{\partial}{\partial x_i}\otimes\frac{\partial}{\partial x_j}$ iff

$$(\nabla\phi)J(\nabla\phi)^{\mathrm{T}}=J. \tag{8}$$

Differentiating (7), we obtain,

$$dq_{k+1}=e^{Ah}dq_k+h\cdot e^{A\frac{h}{2}}\nabla f\left(\frac{1}{2}e^{A\frac{h}{2}}dq_k+\frac{1}{2}e^{-A\frac{h}{2}}dq_{k+1}\right),$$
$$\left(I-\frac{h}{2}\cdot e^{A\frac{h}{2}}\nabla f e^{-A\frac{h}{2}}\right)dq_{k+1}=\left(e^{Ah}+\frac{h}{2}\cdot e^{A\frac{h}{2}}\nabla f e^{A\frac{h}{2}}\right)dq_k.$$

So the map $\phi(q_k)=q_{k+1}$ has Jacobian matrix $\nabla\phi=M^{-1}N$, where

$$M=I-\frac{h}{2}\cdot e^{A\frac{h}{2}}\nabla f e^{-A\frac{h}{2}}=I-\frac{h}{2}\cdot e^{A\frac{h}{2}}J\nabla^2 V e^{-A\frac{h}{2}},$$
$$N=e^{Ah}+\frac{h}{2}\cdot e^{A\frac{h}{2}}\nabla f e^{A\frac{h}{2}}=e^{Ah}+\frac{h}{2}\cdot e^{A\frac{h}{2}}J\nabla^2 V e^{A\frac{h}{2}}.$$

Then, we just need to verify (8), which is equivalent to $MJM^{\mathrm{T}}=NJN^{\mathrm{T}}$,

$$\begin{aligned}MJM^{\mathrm{T}}&=\left(I-\frac{h}{2}\cdot e^{A\frac{h}{2}}J\nabla^2 V e^{-A\frac{h}{2}}\right)J\left(I+\frac{h}{2}\cdot e^{A\frac{h}{2}}\nabla^2 V J e^{-A\frac{h}{2}}\right)\\&=J-\frac{h^2}{4}e^{A\frac{h}{2}}J\nabla^2 V e^{-A\frac{h}{2}}Je^{A\frac{h}{2}}\nabla^2 V J e^{-A\frac{h}{2}}\\&=J-\frac{h^2}{4}e^{A\frac{h}{2}}J\nabla^2 V J\nabla^2 V J e^{-A\frac{h}{2}}\\&=NJN^{\mathrm{T}}.\end{aligned} \tag{9}$$

In (9), we used the property that $\nabla^2 V$ is symmetric, and that A is skew-symmetric which implies that $(e^{Ah})^{\mathrm{T}}=e^{-Ah}$. Furthermore, the assumption that D and J commutes, implies that the matrix exponential $e^{A\frac{h}{2}}$, where $A=JD$, also commutes with J. □

The exponential midpoint rule is a second-order method, and we will now develop higher-order symplectic exponential integrators. Recall that a general diagonally implicit Symplectic Runge–Kutta method (DISRK) [4, Theorem 4.4] has a Butcher tableau of the form given in Table 1.

If we apply the DISRK method to the transformed system (2), and then convert back, we obtain the following diagonally implicit symplectic exponential (DISEX) integrator,

$$\begin{cases}Q_i=e^{Ahc_i}q_k+h\cdot\sum_{j=1}^{i}a_{ij}e^{Ah(c_i-c_j)}f(Q_j),\\ q_{k+1}=e^{Ah}q_k+h\cdot\sum_{i=1}^{s}b_i e^{Ah(1-c_i)}f(Q_i),\end{cases} \tag{10}$$

Table 1
DISRK.

c_1	$\frac{b_1}{2}$	0	0	0	0
c_2	b_1	$\frac{b_2}{2}$	0	0	0
c_3	b_1	b_2	$\frac{b_3}{2}$	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_s	b_1	b_2	b_3	$\cdots$	$\frac{b_s}{2}$
	b_1	b_2	b_3	$\cdots$	b_s

Table 2
DISEX.

e^{Ahc_1}	$\frac{b_1}{2}$	0	0	0	0
e^{Ahc_2}	$b_1 e^{Ah(c_2-c_1)}$	$\frac{b_2}{2}$	0	0	0
e^{Ahc_3}	$b_1 e^{Ah(c_3-c_1)}$	$b_2 e^{Ah(c_3-c_2)}$	$\frac{b_3}{2}$	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
e^{Ahc_s}	$b_1 e^{Ah(c_s-c_1)}$	$b_2 e^{Ah(c_s-c_2)}$	$b_3 e^{Ah(c_s-c_3)}$	$\cdots$	$\frac{b_s}{2}$
e^{Ah}	$b_1 e^{Ah(1-c_1)}$	$b_2 e^{Ah(1-c_2)}$	$b_3 e^{Ah(1-c_3)}$	$\cdots$	$b_s e^{Ah(1-c_s)}$

where a_{ij} are the coefficients in Table 1. This integrator can be represented in terms of the Butcher tableau given in Table 2.

It was shown in [15] that any DISRK method can be regarded as the composition of midpoint rules with timesteps $b_1h, b_2h, b_3h, \cdots b_sh$. As such, it is natural to ask whether the DISEX method is also the composition of exponential midpoint rules. Notice that (1) is autonomous while the transformed system (2) is nonautonomous, and our application of the DISRK method to (2) involves time explicitly, so the result does not immediately follow from the result of [15]. However, the same conclusion still holds for the DISEX method, as demonstrated by the following theorem:

Theorem 2. *The DISEX method is equivalent to the composition of exponential midpoint rules with timesteps $b_1h, b_2h, b_3h, \cdots b_sh$.*

Proof. The composition of exponential midpoint rules

$$q_k \xrightarrow{b_1h} Z_1 \xrightarrow{b_2h} Z_2 \cdots \xrightarrow{b_sh} Z_s = q_{k+1}$$

is represented as follows,

$$Z_1 = e^{Ahb_1} q_k + b_1 h \cdot e^{Ah\frac{b_1}{2}} f\left(\frac{e^{Ah\frac{b_1}{2}} q_k + e^{-Ah\frac{b_1}{2}} Z_1}{2}\right), \tag{M.1}$$

$$Z_2 = e^{Ahb_2} Z_1 + b_2 h \cdot e^{Ah\frac{b_2}{2}} f\left(\frac{e^{Ah\frac{b_2}{2}} Z_1 + e^{-Ah\frac{b_2}{2}} Z_2}{2}\right), \tag{M.2}$$

$$\vdots$$

$$Z_i = e^{Ahb_i} Z_{i-1} + b_i h \cdot e^{Ah\frac{b_i}{2}} f\left(\frac{e^{Ah\frac{b_i}{2}} Z_{i-1} + e^{-Ah\frac{b_i}{2}} Z_i}{2}\right), \tag{M.i}$$

$$\vdots$$

$$Z_s = e^{Ahb_s} Z_{s-1} + b_s h \cdot e^{Ah\frac{b_s}{2}} f\left(\frac{e^{Ah\frac{b_s}{2}} Z_{s-1} + e^{-Ah\frac{b_s}{2}} Z_s}{2}\right). \tag{M.s}$$

Introduce $Q_1 = \frac{e^{Ah\frac{b_1}{2}} q_k + e^{-Ah\frac{b_1}{2}} Z_1}{2}$ in (M.1) as an intermediate variable. Then, on both sides of (M.1), multiply by $e^{-Ah\frac{b_1}{2}}$, add $e^{Ah\frac{b_1}{2}} q_k$, and divide by two, which yields an equivalent form of (M.1),

$$Q_1 = e^{Ah\frac{b_1}{2}} q_k + \frac{b_1}{2} h \cdot f(Q_1). \tag{S.1}$$

For the Runge–Kutta method represented by the Butcher tableau in Table 1 to be consistent, the coefficients have to satisfy,

$$\begin{cases} c_1 = \dfrac{b_1}{2}, \\ c_2 = b_1 + \dfrac{b_2}{2}, \\ \quad \vdots \\ c_i = b_1 + b_2 + \cdots b_{i-1} + \dfrac{b_i}{2}, \\ \quad \vdots \\ c_s = b_1 + b_2 + b_3 + \cdots + b_{s-1} + \dfrac{b_s}{2}, \\ 1 = b_1 + b_2 + b_3 + \cdots + b_{s-1} + b_s. \end{cases}$$

So equation (S.1) coincides with the first line of the Butcher tableau of the DISEX method. Similarly, introduce $Q_2 = \frac{e^{Ah\frac{b_2}{2}}Z_1 + e^{-Ah\frac{b_2}{2}}Z_2}{2}$. Then, on both sides of (M.2), multiply by $e^{-Ah\frac{b_2}{2}}$, add $e^{Ah\frac{b_2}{2}}Z_1$, then divided by two, which yields an equivalent form of (M.2):

$$\begin{aligned} Q_2 &= e^{Ah\frac{b_2}{2}}Z_1 + \frac{b_2}{2}h\cdot f(Q_2) \\ &= e^{Ah(b_1+\frac{b_2}{2})}q_k + b_1 h\cdot e^{Ah(\frac{b_1}{2}+\frac{b_2}{2})}f(Q_1) + \frac{b_2}{2}h\cdot f(Q_2) \\ &= e^{Ahc_2}q_k + b_1 h\cdot e^{Ah(c_2-c_1)}f(Q_1) + \frac{b_2}{2}h\cdot f(Q_2). \end{aligned} \tag{S.2}$$

So equation (S.2) coincides with the second line of the Butcher tableau of the DISEX method. Then, as before, we introduce $Q_i = \frac{e^{Ah\frac{b_i}{2}}Z_{i-1} + e^{-Ah\frac{b_i}{2}}Z_i}{2}$, and apply the same technique to (M.i), which yields

$$Q_i = e^{Ah\frac{b_i}{2}}Z_{i-1} + \frac{b_i}{2}h\cdot f(Q_i).$$

By induction,

$$\begin{aligned} Z_{i-1} = e^{Ah(b_{i-1}+\cdots+b_2+b_1)}q_k &+ b_1 h\cdot e^{Ah(b_{i-1}+b_{i-2}+\cdots+\frac{b_1}{2})}f(Q_1) \\ &+ b_2 h\cdot e^{Ah(b_{i-1}+b_{i-2}+\cdots+\frac{b_2}{2})}f(Q_2) + \cdots b_{i-1}h\cdot e^{Ah(\frac{b_{i-1}}{2})}f(Q_{i-1}), \end{aligned}$$

so

$$\begin{aligned} Q_i = {} & e^{Ah(\frac{b_i}{2}+b_{i-1}+\cdots+b_2+b_1)}q_k + b_1 h\cdot e^{Ah(\frac{b_i}{2}+b_{i-1}+b_{i-2}+\cdots+\frac{b_1}{2})}f(Q_1) \\ & + b_2 h\cdot e^{Ah(\frac{b_i}{2}+b_{i-1}+b_{i-2}+\cdots+\frac{b_2}{2})}f(Q_2) + \cdots \\ & + b_{i-1}h\cdot e^{Ah(\frac{b_i}{2}+\frac{b_{i-1}}{2})}f(Q_{i-1}) + \frac{b_i}{2}h\cdot f(Q_i) \\ = {} & e^{Ahc_i}q_k + b_1 h\cdot e^{Ah(c_i-c_1)}f(Q_1) + \cdots \\ & + b_{i-1}h\cdot e^{Ah(c_i-c_{i-1})}f(Q_{i-1}) + \frac{b_i}{2}h\cdot f(Q_i), \end{aligned} \tag{S.i}$$

which coincides with the i-th row of the Butcher tableau of the DISEX method. Finally, we have

$$\begin{aligned} q_{k+1} &= Z_s \\ &= e^{Ahb_s}Z_{s-1} + b_s h\cdot f(Q_s) \\ &= e^{Ah(b_s+\cdots b_2+b_1)}q_k + b_1 h\cdot e^{Ah(b_s+\cdots b_2+\frac{b_1}{2})}f(Q_1) + \cdots \\ &\quad + b_{s-1}h\cdot e^{Ah(b_s+\frac{b_{s-1}}{2})}f(Q_{s-1}) + b_s h\cdot e^{Ah\frac{b_s}{2}}f(Q_s) \\ &= e^{Ah}q_k + b_1 h\cdot e^{Ah(1-c_1)}f(Q_1) + \cdots b_s h\cdot e^{Ah(1-c_s)}f(Q_s), \end{aligned}$$

which coincides with the last row of the Butcher tableau of the DISEX method. So the composition of exponential midpoint rules with timesteps $b_1h, b_2h, b_3h, \cdots b_sh$ is equivalent to the DISEX method of Table 2. □

Theorem 2 establishes an equivalent relationship between the DISEX method and exponential midpoint rules on the nonautonomous system (2) as was established in [15] for the DISRK method and midpoint rules for the autonomous system (1), and it also shows that the DISEX method preserves the Poisson structure.

3. Energy-preserving exponential integrator

Though classical symplectic methods exhibit superior long time stability, it was observed that symplectic schemes are less competitive for the numerical integration of stiff systems with high frequency. In sharp contrast, energy-preserving methods perform much better [12]. A general way to construct an energy-preserving method for a Poisson system $\dot{q} = J\nabla H(q)$ is the discrete gradient method [11]. We design a discrete gradient $\overline{\nabla} H(q_k, q_{k+1})$ that satisfies the following property,

$$\overline{\nabla} H(q_k, q_{k+1}) \cdot (q_{k+1} - q_k) = H(q_{k+1}) - H(q_k). \tag{11}$$

Then, the resulting discrete gradient method is given by,

$$\frac{q_{k+1} - q_k}{h} = J\overline{\nabla} H(q_k, q_{k+1}). \tag{12}$$

Multiplying $\overline{\nabla} H(q_k, q_{k+1})$ on both sides of (12), we obtain

$$\begin{aligned} H(q_{k+1}) - H(q_k) &= \overline{\nabla} H(q_k, q_{k+1}) \cdot (q_{k+1} - q_k) \\ &= h \cdot \overline{\nabla} H(q_k, q_{k+1}) J \overline{\nabla} H(q_k, q_{k+1}) \\ &= 0. \end{aligned} \tag{13}$$

The last equation of (13) holds simply due to the skew-symmetric property of matrix J, which implies that discrete gradient method (12) preserves energy. We shall combine exponential integrators with the discrete gradient method to obtain an energy-preserving exponential integrator. This approach was initially proposed in [14] for separable Hamiltonian systems using the extended discrete gradient method, and we generalize this to semilinear Poisson systems. Replace the $f(q_k)$ term in the exponential Euler method (4) by the discrete gradient $J\overline{\nabla} V(q_k, q_{k+1})$, which yields

$$q_{k+1} = e^{Ah} q_k + \int_0^h e^{A\tau} d\tau \cdot J\overline{\nabla} V(q_k, q_{k+1}). \tag{14}$$

Theorem 3. *Method (14) preserves the Hamiltonian $H(q)$.*

Proof. Let $S = e^{Ah}$, $T = \int_0^h e^{A\tau} d\tau$, then $q_{k+1} = Sq_k + TJ\overline{\nabla} V(q_k, q_{k+1})$. It is beneficial to observe that the assumption that J and D commute implies that the matrices J, D, A, S, and T all commute with each other. Also, we will show that the following properties hold:

(1) $S^{\mathrm{T}} = S^{-1}$;
(2) $AT = S - I$;
(3) $AT^{\mathrm{T}} = I - S^{\mathrm{T}}$;
(4) $S^{\mathrm{T}} T = T^{\mathrm{T}}$.

Property 1 follows from the fact that $S^{\mathrm{T}} = (e^{Ah})^{\mathrm{T}} = e^{-Ah} = S^{-1}$. Property 2 follows from

$$e^{Ah} - I = e^{A\tau}\Big|_0^h = \int_0^h A \cdot e^{A\tau} d\tau = AT.$$

Taking transposes on both sides of Property 2 and using the fact that A and T commute gives Property 3. Property 4 follow from

$$S^{\mathrm{T}} T = e^{-Ah} \int_0^h e^{A\tau} d\tau = \int_0^h e^{-A(h-\tau)} d\tau = \int_0^h e^{-A\tau} d\tau = T^{\mathrm{T}}.$$

From this, we obtain

$$\begin{aligned}\frac{1}{2}q_{k+1}^{\mathrm{T}}Dq_{k+1} &= \frac{1}{2}q_{k+1}^{\mathrm{T}}D(Sq_k + TJ\overline{\nabla}V)\\ &= \frac{1}{2}(Sq_k + TJ\overline{\nabla}V)^{\mathrm{T}}DSq_k + \frac{1}{2}q_{k+1}^{\mathrm{T}}DTJ\overline{\nabla}V\\ &= \frac{1}{2}q_k^{\mathrm{T}}S^{\mathrm{T}}DSq_k + \frac{1}{2}q_k^{\mathrm{T}}(I - S^{\mathrm{T}})\overline{\nabla}V + \frac{1}{2}q_{k+1}^{\mathrm{T}}(S - I)\overline{\nabla}V\\ &= \frac{1}{2}q_k^{\mathrm{T}}Dq_k - (q_{k+1} - q_k)^{\mathrm{T}}\overline{\nabla}V - \frac{1}{2}q_k^{\mathrm{T}}(S^{\mathrm{T}} + I)\overline{\nabla}V + \frac{1}{2}q_{k+1}^{\mathrm{T}}(S + I)\overline{\nabla}V\\ &= \frac{1}{2}q_k^{\mathrm{T}}Dq_k - V(q_{k+1}) + V(q_k) + \frac{1}{2}\overline{\nabla}V^{\mathrm{T}}J^{\mathrm{T}}T^{\mathrm{T}}(S + I)\overline{\nabla}V\\ &= \frac{1}{2}q_k^{\mathrm{T}}Dq_k - V(q_{k+1}) + V(q_k).\end{aligned} \tag{15}$$

Here, we have used the fact that $DTJ = S - I$, $S^{\mathrm{T}}DS = D$, $S^{\mathrm{T}}DTJ = I - S^{\mathrm{T}}$, and $J^{\mathrm{T}}T^{\mathrm{T}}(S + I) = J^{\mathrm{T}}T - JT^{\mathrm{T}}$ which is skew-symmetric. The above calculation demonstrates that $H(q_{k+1}) = H(q_k)$. □

4. Numerical methods

We will apply our proposed geometric exponential integrators to the semi-discretization of two Hamiltonian PDEs, as the resulting system of differential equations can be expressed in the semilinear form (1), with a coefficient matrix A whose spectral radius increases with the spatial resolution of the discretization. The first is the nonlinear Schrödinger equation,

$$i\psi_t + \psi_{xx} - 2|\psi|^2\psi = 0, \tag{16}$$

in which $\psi = u + iv$ is the wave function with real part u and imaginary part v, and has the following equivalent form,

$$\begin{cases} u_t = -v_{xx} + 2(u^2 + v^2)v,\\ v_t = u_{xx} - 2(u^2 + v^2)u. \end{cases} \tag{17}$$

The second is the KdV equation,

$$u_t + uu_x + u_{xxx} = 0. \tag{18}$$

To discretize the two PDEs, we impose 2π periodic boundary conditions. Given a smooth 2π periodic function $f(x)$, on the interval $[0, 2\pi]$, choose $2n+1$ equispaced interpolation points $x_j = jh$, $j = 0, 1, 2, \ldots, 2n$, $h = \frac{2\pi}{2n+1}$. Given nodal values $\{v_j\}_{j=0}^{2n}$, there exists a unique trigonometric polynomial $v(x)$ with degree less or equal n, such that, $v(x_j) = v_j$ (see, for example, [1]).

$$v(x) = \sum_{k=-n}^{n} \hat{v}_k e^{ikx}, \quad \hat{v}_k = \frac{1}{2n+1}\sum_{j=0}^{2n} v_j e^{-ikx_j}.$$

By substituting the expression for the coefficients $\hat{v}_k$, we obtain

$$v(x) = \sum_{j=0}^{2n} v_j \phi(x - x_j) = \sum_{j=0}^{2n} v_j \phi_j(x), \tag{19}$$

where

$$\phi(x) = \sum_{k=-n}^{n} \frac{1}{2n+1} e^{ikx} = \frac{1}{2n+1}\frac{\sin((n+\frac{1}{2})x)}{\sin(\frac{x}{2})}.$$

From this, we see that $\{e^{ikx}\}_{k=-n}^{n}$ and $\{\phi_j\}_{j=0}^{2n}$ are equivalent orthogonal bases for the trigonometric polynomial function space, and each such function can be parametrized by either the nodal values $\{v_j\}_{j=0}^{2n}$ or Fourier coefficients $\{\hat{v}_k\}_{k=-n}^{n}$. They represent the same function, but with respect to two different bases. The transformation between $\{v_j\}_{j=0}^{2n}$ and $\{\hat{v}_k\}_{k=-n}^{n}$ can be performed using the Fast Fourier transformation (FFT), which has $O(n \log n)$ complexity.

The first and second-order differentiation matrices [13] with respect to the representation in terms of nodal values $\{v_j\}_{j=0}^{2n}$ are given by

$$(D_1)_{kj} = \begin{cases} 0, & k = j, \\ \frac{(-1)^{(k-j)}}{2\sin(\frac{(k-j)h}{2})}, & k \neq j, \end{cases}$$

$$(D_2)_{kj} = \begin{cases} -\frac{n(n+1)}{3}, & k = j, \\ \frac{(-1)^{(k-j+1)}\cos(\frac{(k-j)h}{2})}{2\sin^2(\frac{(k-j)h}{2})}, & k \neq j, \end{cases}$$

respectively. However, with respect to the representation in terms of Fourier coefficients $\{\hat{v}_k\}_{k=-n}^{n}$, they are diagonal,

$$\hat{D}_1 = \mathrm{diag}(ik)_{k=-n}^{n}, \quad \hat{D}_2 = \mathrm{diag}(-k^2)_{k=-n}^{n}.$$

We can also define a third-order differentiation matrix D_3, which has the property $D_3 = D_1 D_2 = D_2 D_1$, and it is diagonal with respect to the Fourier coefficients $\hat{D}_3 = \mathrm{diag}(-ik^3)_{k=-n}^{n}$. The observation that D_1, D_2 and D_3 are all diagonal in the Fourier representation is critical to a fast implementation of the product of matrix functions with vectors.

Theorem 4. *Suppose that D_1, D_2, D_3 are the first, second and third-order differentiation matrices, respectively, $q = \{q_j\}_{j=0}^{2n}$ is a vector with Fourier transform $F[q] = \hat{q} = \{\hat{q}_k\}_{k=-n}^{n}$, and f is an analytic function. Then,*

$$f(D_1)q = F^{-1}[\mathrm{diag}(f(ik))\hat{q}], \qquad f(D_2)q = F^{-1}[\mathrm{diag}(f(-k^2))\hat{q}], \qquad f(D_3)q = F^{-1}[\mathrm{diag}(f(-ik^3))\hat{q}],$$

where F^{-1} is the inverse Fourier transform.

Proof. Recall that the matrix D_2 is diagonalizable with eigenvalues $\lambda_k = -k^2$, and corresponding eigenvectors $e_k = \{e^{ikx_j}\}_{j=0}^{2n}$,

$$f(D_2)q = f(D_2)\left(\sum_{k=-n}^{n} \hat{q}_k \cdot e_k\right) = \sum_{k=-n}^{n} \hat{q}_k \cdot f(D_2)e_k = \sum_{k=-n}^{n} \hat{q}_k \cdot f(\lambda_k)e_k = F^{-1}[\mathrm{diag}(f(\lambda_k))\hat{q}] = F^{-1}[\mathrm{diag}(f(-k^2))\hat{q}].$$

Notice that D_1 is also diagonalizable with eigenvalues $\lambda_k = ik$, and corresponding eigenvectors e_k, D_3 is diagonalizable with eigenvalues $\lambda_k = -ik^3$, and corresponding eigenvectors e_k, so the property that $f(D_1)q = F^{-1}[\mathrm{diag}(f(ik))\hat{q}]$, $f(D_3)q = F^{-1}[\mathrm{diag}(f(-ik^3))\hat{q}]$ can be verified in the same way. □

4.1. Nonlinear Schrödinger equation

We perform a semi-discretization of (17) by discretizing the solution u, v in space using their corresponding nodal values $\{q_j\}$ and $\{p_j\}$. Applying the pseudospectral method, we obtain the following system of ODEs,

$$\begin{cases} \dot{q} = -D_2 p + 2(q^2 + p^2)p, \\ \dot{p} = D_2 q - 2(q^2 + p^2)q, \end{cases} \tag{20}$$

where the nonlinear term $(q^2 + p^2)p$ is computed elementwise, and represents the vector consisting of $\{(q_j^2 + p_j^2)p_j\}$ entries. We adopt this notation throughout the rest of the paper for brevity. Then, (20) can be expressed as,

$$\frac{d}{dt}\begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix}\begin{pmatrix} q \\ p \end{pmatrix} + \begin{pmatrix} 2(q^2 + p^2)p \\ -2(q^2 + p^2)q \end{pmatrix}, \tag{21}$$

where

$$A = \begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}\begin{pmatrix} -D_2 & 0 \\ 0 & -D_2 \end{pmatrix} = J \cdot D,$$

$$f(q,p) = \begin{pmatrix} 2(q^2 + p^2)p \\ -2(q^2 + p^2)q \end{pmatrix} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}\begin{pmatrix} 2(q^2 + p^2)q \\ 2(q^2 + p^2)p \end{pmatrix} = J \cdot \nabla V(q,p),$$

and $V(q,p) = \frac{1}{2}(q^2 + p^2)^2$. It is easy to verify that J is skew-symmetric, D is symmetric, and $JD = DJ$. Thus, (21) is a semilinear Poisson system.

To apply the exponential midpoint rule, we need to compute the product of a matrix function and a vector, which has the form $e^{Ah}\begin{pmatrix} q \\ p \end{pmatrix}$,

$$e^{\begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix}} = \sum_{k=0}^{\infty} \frac{1}{k!} \begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix}^k = \begin{pmatrix} \cos(D_2) & -\sin(D_2) \\ \sin(D_2) & \cos(D_2) \end{pmatrix}.$$

By Theorem 4, we have that

$$\begin{aligned} e^{\begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix} h} \begin{pmatrix} q \\ p \end{pmatrix} &= \begin{pmatrix} \cos(D_2 h) & -\sin(D_2 h) \\ \sin(D_2 h) & \cos(D_2 h) \end{pmatrix} \begin{pmatrix} q \\ p \end{pmatrix} \\ &= \begin{pmatrix} \cos(D_2 h) \cdot q - \sin(D_2 h) \cdot p \\ \sin(D_2 h) \cdot q + \cos(D_2 h) \cdot p \end{pmatrix} \\ &= \begin{pmatrix} F^{-1}[\cos(k^2 h)\hat{q}_k + \sin(k^2 h)\hat{p}_k] \\ F^{-1}[\cos(k^2 h)\hat{p}_k - \sin(k^2 h)\hat{q}_k] \end{pmatrix}. \end{aligned} \tag{22}$$

In summary, the exponential midpoint rule for the nonlinear Schrödinger equation is given by

$$z_{k+1} = e^{Ah} z_k + h \cdot e^{A\frac{h}{2}} f\left(\frac{e^{A\frac{h}{2}} z_k + e^{-A\frac{h}{2}} z_{k+1}}{2} \right),$$

where $z_k = \begin{pmatrix} q_k \\ p_k \end{pmatrix}$, $z_{k+1} = \begin{pmatrix} q_{k+1} \\ p_{k+1} \end{pmatrix}$, $A = \begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix}$, and $e^{Ah} z_k$ can be efficiently calculated using (22).

For the energy-preserving exponential integrator for the nonlinear Schrödinger equation,

$$z_{k+1} = e^{Ah} z_k + \int_0^h e^{A\tau} d\tau \cdot J \overline{\nabla} V(z_k, z_{k+1}).$$

Here,

$$\begin{aligned} \int_0^h e^{\begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix} \tau} d\tau &= \int_0^h \begin{pmatrix} \cos(D_2\tau) & -\sin(D_2\tau) \\ \sin(D_2\tau) & \cos(D_2\tau) \end{pmatrix} d\tau \\ &= h \cdot \begin{pmatrix} \frac{\sin(D_2 h)}{D_2 h} & \frac{\cos(D_2 h)-1}{D_2 h} \\ \frac{1-\cos(D_2 h)}{D_2 h} & \frac{\sin(D_2 h)}{D_2 h} \end{pmatrix}, \end{aligned}$$

and by Theorem 4,

$$\begin{aligned} \int_0^h e^{\begin{pmatrix} 0 & -D_2 \\ D_2 & 0 \end{pmatrix} \tau} d\tau \cdot \begin{pmatrix} q \\ p \end{pmatrix} &= \begin{pmatrix} \frac{\sin(D_2 h)}{D_2} q + \frac{\cos(D_2 h)-1}{D_2} p \\ \frac{1-\cos(D_2 h)}{D_2} q + \frac{\sin(D_2 h)}{D_2} p \end{pmatrix}, \\ &= \begin{pmatrix} F^{-1}[\frac{\sin(k^2 h)}{k^2}\hat{q}_k - \frac{\cos(k^2 h)-1}{k^2}\hat{p}_k] \\ F^{-1}[-\frac{1-\cos(k^2 h)}{k^2}\hat{q}_k + \frac{\sin(k^2 h)}{k^2}\hat{p}_k] \end{pmatrix}. \end{aligned}$$

We can construct the discrete gradient

$$\overline{\nabla} V(z_k, z_{k+1}) = \begin{pmatrix} 2({q^2}_{k+\frac{1}{2}} + {p^2}_{k+\frac{1}{2}}) \cdot q_{k+\frac{1}{2}} \\ 2({q^2}_{k+\frac{1}{2}} + {p^2}_{k+\frac{1}{2}}) \cdot p_{k+\frac{1}{2}} \end{pmatrix},$$

where

$$q_{k+\frac{1}{2}} = \frac{q_k + q_{k+1}}{2}, \quad p_{k+\frac{1}{2}} = \frac{p_k + p_{k+1}}{2}, \tag{23}$$

$${q^2}_{k+\frac{1}{2}} = \frac{q_k^2 + q_{k+1}^2}{2}, \quad {p^2}_{k+\frac{1}{2}} = \frac{p_k^2 + p_{k+1}^2}{2}. \tag{24}$$

Notice that $\overline{\nabla} V(z_k, z_{k+1})$ is symmetric with respect to z_k and z_{k+1}, and it can be verified that it satisfies (11). A classical discrete gradient method can be constructed as follows,

$$z_{k+1} = z_k + h J \overline{\nabla} H(z_k, z_{k+1}), \tag{25}$$

where

$$\overline{\nabla} H(z_k, z_{k+1}) = \begin{pmatrix} -D_2 & 0 \\ 0 & -D_2 \end{pmatrix} \frac{z_k + z_{k+1}}{2} + \overline{\nabla} V(z_k, z_{k+1}).$$

The method described by (25) is very similar to classical midpoint rule, the only difference is that in $\overline{\nabla} V(z_k, z_{k+1})$, ${q^2}_{k+\frac{1}{2}}$ is used, while $(q_{k+\frac{1}{2}})^2$ is used in the midpoint rule, so (25) can be viewed as a modified midpoint rule.

4.2. KdV equation

Rewrite (18) as

$$u_t = \Big(-\frac{\partial}{\partial x}\Big)\Big(\frac{1}{2}u^2 + u_{xx}\Big),$$

then apply pseudospectral semi-discretization to obtain the following system,

$$\begin{aligned} \dot{q} &= (-D_1)\Big(\frac{1}{2}q^2 + D_2 q\Big) \\ &= (-D_1)D_2 q + (-D_1)\Big(\frac{1}{2}q^2\Big), \end{aligned}$$

which has the form of a semilinear Poisson system (5),

$$\dot{q} = J(Dq + \nabla V(q)) = J\nabla H(q),$$

where $J = -D_1$, $D = D_2$, $A = JD = -D_3$, $\nabla V(q) = \frac{1}{2}q^2$, $H(q) = \frac{1}{2}q^{\mathrm{T}} D_2 q + \frac{1}{6}q^3$. The exponential midpoint rule for KdV reads as follows,

$$q_{k+1} = e^{-D_3 h} q_k + h \cdot e^{-D_3 \frac{h}{2}} f\Big(\frac{e^{-D_3 \frac{h}{2}} q_k + e^{D_3 \frac{h}{2}} q_{k+1}}{2}\Big),$$

and the energy preserving exponential integrator is given by,

$$q_{k+1} = e^{-D_3 h} q_k + \int_0^h e^{-D_3 \tau} d\tau \cdot (-D_1)\overline{\nabla} V(q_k, q_{k+1}),$$

with discrete gradient $\overline{\nabla} V(q_k, q_{k+1}) = \frac{1}{6}(q_k^2 + q_k \cdot q_{k+1} + q_{k+1}^2)$. A related classical discrete gradient method can be constructed as follows,

$$\overline{\nabla} H(q_k, q_{k+1}) = (D_2)\frac{q_k + q_{k+1}}{2} + \overline{\nabla} V(q_k, q_{k+1}). \tag{26}$$

By Theorem 4, in each iteration, the matrix function and vector product can be implemented as

$$e^{-D_3 h} q = F^{-1}[e^{ik^3 h}\hat{q}_k],$$

and

$$\Big(\int_0^h e^{-D_3 \tau} d\tau\Big) q = \Big(\frac{e^{-D_3 h} - I}{-D_3}\Big) q = F^{-1}\Big[\frac{e^{ik^3 h} - 1}{ik^3}\hat{q}_k\Big].$$

4.3. Remarks

We have not analyzed the stiff order and long-time behavior of the methods proposed in this paper. In particular, it should be observed that the standard backward error analysis results for symplectic integration of Hamiltonian systems do not apply to the highly oscillatory systems that we are considering here. A rigorous proof of long-time energy stability will likely involve modulated Fourier expansions, as in [2,3].

Table 3
Maximum timestep and average iteration number for convergence for the nonlinear Schrödinger equation, as a function of the numerical integrator, nonlinear solver, and spatial resolution.

n	Midpoint				Midpoint exp		Discrete gradient		Energy exp		DISEX	
	Fixed point		Newton		Fixed point		Fixed point		Fixed point		Fixed point	
	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$
11	0.02	14.2	0.1	8.3	0.1	12.6	0.02	14	0.1	11.9	0.1	101.4
21	0.01	12.9	0.1	8.3	0.08	12.6	0.01	11.8	0.1	12	0.03	47.7
41	4×10^{-3}	7.9	0.1	8.3	0.06	10.9	4×10^{-3}	5.8	0.1	12	0.03	49.2
61	2×10^{-3}	2.1	0.1	8.3	0.04	8.1	2×10^{-3}	4	0.1	12	0.025	44.5
81	10^{-3}	4.1	0.1	8.3	0.04	8.1	10^{-3}	3	0.1	12	0.025	44.6
121	5×10^{-4}	13.9	0.1	8.3	0.04	8.1	5×10^{-4}	3.4	0.1	12	0.01	43.3
161	2×10^{-4}	3.9	0.1	8.3	0.01	5	2×10^{-4}	2	0.1	12	0.01	30.5
201	1×10^{-4}	3.08	0.1	8.3	8×10^{-3}	4.7	1×10^{-4}	2	0.1	12	8×10^{-3}	30
401	4×10^{-5}	2	0.1	8.3	5×10^{-3}	4.9	4×10^{-5}	1	0.1	12	4×10^{-3}	32

5. Numerical experiments

5.1. Nonlinear Schrödinger equation

In Table 3 above, n denotes the number of nodes we discretize the spatial domain with, and we tabulate the maximum timestep that the nonlinear solver converges for, and the number of iterations taken to converge when averaged over the first thousand timesteps. We consider the convergence properties of fixed point iteration and Newton type iteration for the midpoint rule, and fixed point iteration for the exponential midpoint rule, discrete gradient method (25), energy-preserving exponential integrator, and diagonally implicit symplectic exponential (DISEX) integrator with six stages [7]:

$$b_1 = 0.5080048194000274 \qquad b_2 = 1.360107162294827 \qquad b_3 = 2.019293359181722$$
$$b_4 = 0.5685658926458250 \qquad b_5 = -1.459852049586439 \qquad b_6 = -1.996119183935963.$$

The underlying symplectic diagonally implicit Runge–Kutta method has algebraic order 5, and due to the conjugacy between (1) and (2), the corresponding DISEX integrator also has order 5. The midpoint exponential integrator, discrete gradient integrator and energy preserving exponential integrator, are all implemented using fixed point iteration, while DISEX is implemented as the composition of midpoint exponential integrators. For the Newton iteration, since each iteration requires the LU factorization of the Jacobian, which is computationally prohibitive, we instead implemented a Quasi–Newton method, i.e., the Jacobian of the initial point is used repeatedly until convergence, so the LU factorization need only be computed once per timestep. Since for numerical integration of ODEs, the initial point is well approximated by a high order explicit method, this technique works well in practice.

If the timestep is too large, the nonlinear solver fails to converge, the number of required iterations diverges, or the numerical accuracy of the solution degrades dramatically. Here, we have chosen the maximum timestep by considering both the long time stability and the number of iterations necessary. Since DISEX is implemented as the composition of exponential midpoint rules, the number of iterations reported is the sum of number of iterations for each component exponential midpoint rule.

We observe that when the midpoint rule is implemented using fixed point iterations, there is a significant decay in the allowable timestep as the spatial resolution is increased, whereas the Newton type iteration allows a relatively large timestep that is independent of the spatial resolution. In contrast, the midpoint exponential method exhibits a slower rate of decrease in allowable timestep when using fixed point iterations. When using fixed point iterations, the allowable timestep of the discrete gradient method, which is an energy preserving method, behaves similarly to the midpoint rule, and in contrast, the energy preserving exponential integrator has an allowable timestep that is independent of the spatial resolution.

In Fig. 1, we observe that the allowable timestep when using fixed point iteration scale like n^{-2} for the classical midpoint rule, and like n^{-1} for both the midpoint exponential rule and DISEX. As shown in Fig. 2, the exponential midpoint rule exhibits an energy error that remains small and bounded, which is consistent with it being a symplectic integrator, and the trajectory error grows linearly.

The energy preserving exponential integrator is designed to preserve energy exactly, so even when the timestep is 0.1, we see in Fig. 3 that the energy is still preserved approximately to within machine error. For DISEX, the energy and trajectory error is shown in Fig. 4. Observe that the energy error is small and bounded, as expected of a symplectic integrator, and the trajectory error is small as well, as expected of a higher-order numerical integrator.

In general, the number of iterations decrease as the timestep decreases, which is not surprising since the initial guess is typically more accurate as the timestep decreases. For a fixed time interval, the total computational cost depends on the number of timesteps, average number of iterations per timestep, and the algorithmic complexity of each iteration. To illustrate the relative computational efficiency of the various methods with different solvers, we set $n = 161$, and plot the trajectory error over the time interval $[0, 1]$ vs. total computational cost in Fig. 5. As can be seen from the figure, for

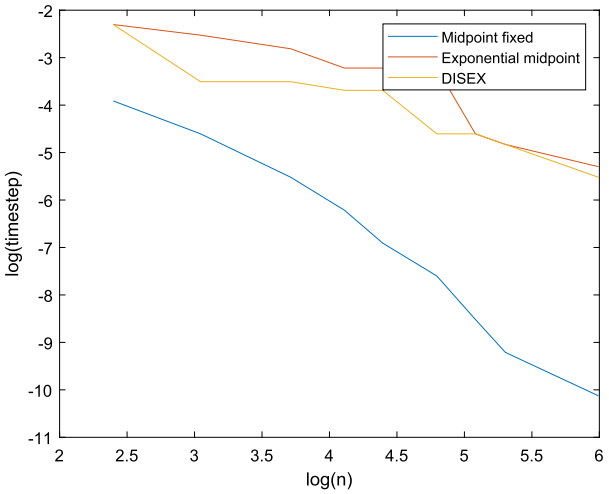
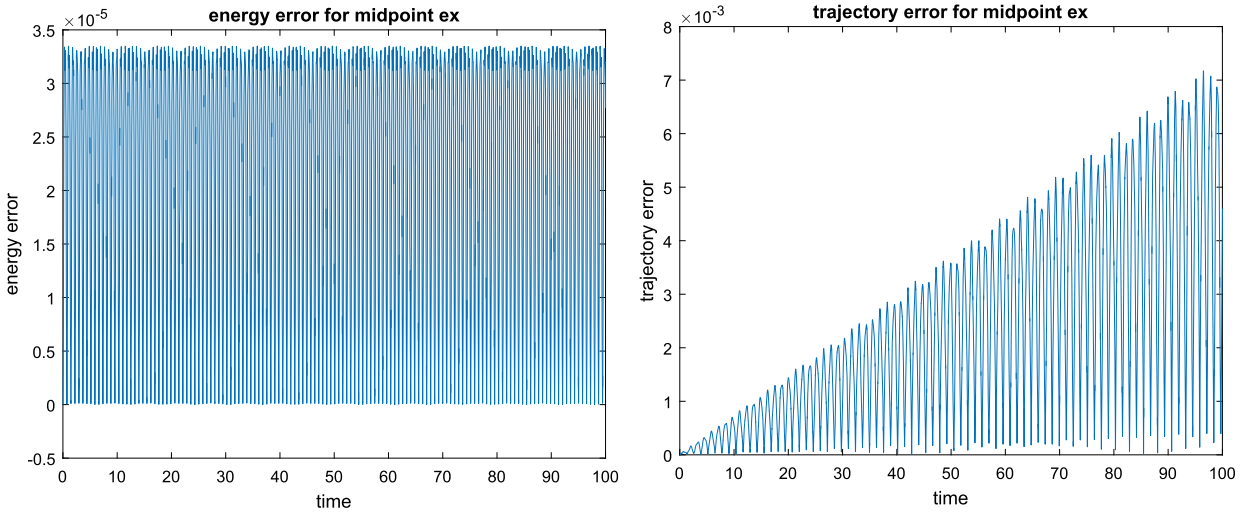


Fig. 1. Maximum timestep for which fixed point iterations converge as a function of the spatial resolution for the nonlinear Schrödinger equation.



(A) Energy error (B) Trajectory error

Fig. 2. Error plots for the exponential midpoint rule applied to the nonlinear Schrödinger equation, $n = 161$, $h = 0.01$.

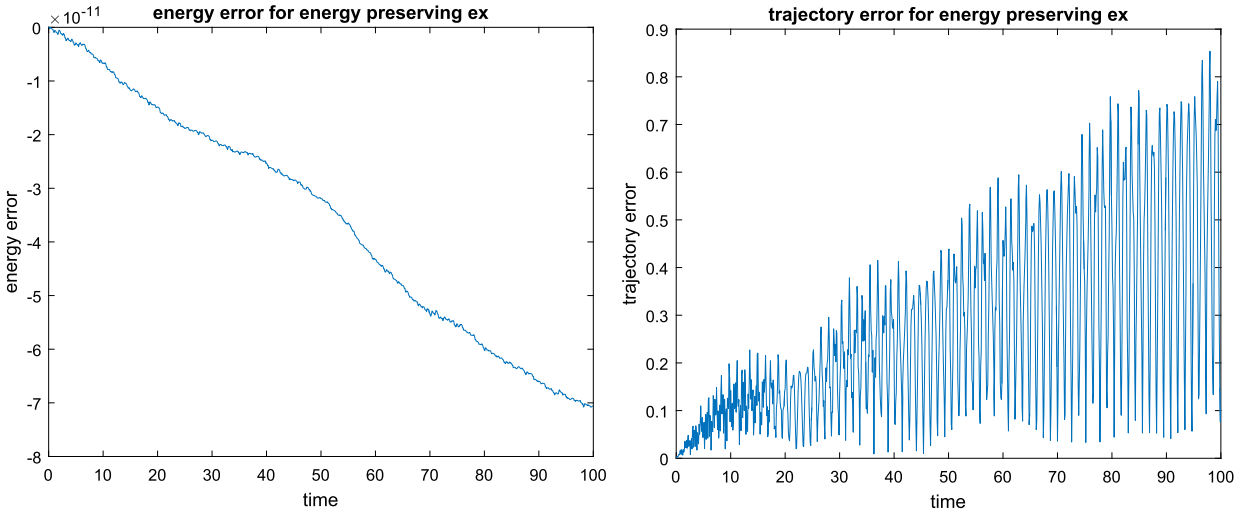
the same trajectory accuracy, the midpoint rule with Newton iterations is the most computationally expensive, due to the higher complexity of the Newton iterations. However, the minimum computational cost achievable using the midpoint rule with fixed point iterations and the discrete gradient methods is rather high, as they require the use of smaller timesteps in order to converge. This problem becomes more severe for non-exponential integrators as the spatial resolution n increases, as the condition number of the matrix A increases with n. In contrast, the minimum computational cost achievable by the midpoint exponential and energy exponential methods is lower, and their trajectory error is also smaller for the same computational cost. For DISEX, the steeper slope indicates that it is a higher-order method, and it is the most economical choice when high accuracy is required.

5.2. KdV

We simulate the KdV equation,

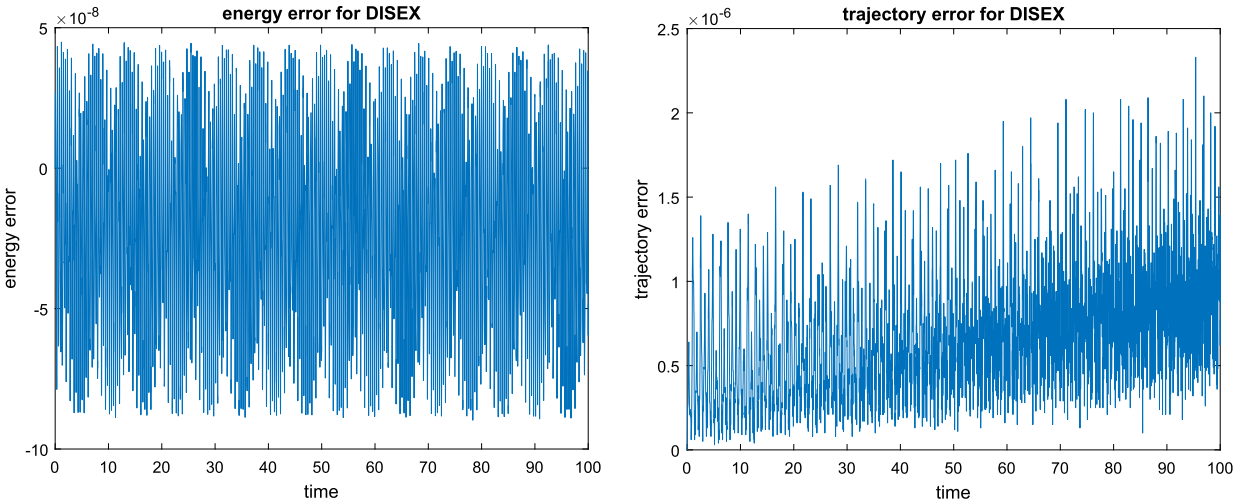
$$u_t + uu_x + \nu u_{xxx} = 0,$$

where $\nu = 5 \times 10^{-4}$. In Table 4, n denotes the number of nodes used to discretize the spatial domain, and we tabulate the maximum timestep for which the nonlinear solver converges, and average number of iterations taken to converge. In particular, we explore the effect of the numerical integrator, the nonlinear solver, and the spatial resolution of the semi-discretization, on the convergence properties of the solver. The maximum timestep is chosen by considering both the long



(A) Energy error (B) Trajectory error

Fig. 3. Error plots for the energy preserving exponential integrator applied to the nonlinear Schrödinger equation, $n = 161$, $h = 0.1$.



(A) Energy error (B) Trajectory error

Fig. 4. Error plots for the 6 stage DISEX applied to the nonlinear Schrödinger equation, $n = 161$, $h = 0.01$.

time stability and the number of iterations necessary, and the number of iterations is averaged over the first one thousand timesteps. For the midpoint rule, the maximum timestep decreases like n^{-3} for fixed point iterations, and a comparable timestep is required for the Newton iterations to converge, which renders it too costly in practice. It is interesting to compare this with the convergence properties of the NLS problem, where the maximum timestep for the midpoint rule with fixed point iteration decreases like n^{-2}. This difference is due to the fact that for NLS, the second order derivative term ψ_{xx} introduces a D_2 term in the semi-discrete ODE, and its spectral radius increases quadratically; whereas for KdV, u_{xxx} introduces a D_3 term in the semi-discrete ODE, and its spectral radius increases cubicly. The classical discrete gradient method for the KdV equation is given by (26), and it exhibits the same timestep restrictions as the midpoint rule. In contrast, both the exponential midpoint and energy preserving exponential integrator allow rather large timesteps that are independent of the spatial resolution. Even though DISEX is implemented as the composition of exponential midpoint rules, it requires smaller timesteps than a single exponential midpoint rule.

In Fig. 6, we observe that the exponential midpoint rule has an energy error that is small and bounded, as is typical for a symplectic integrator, and the trajectory error grows linearly. In Fig. 7, the energy preserving exponential integrator has an energy that is preserved to within machine precision, and the trajectory error grows linearly.

As in the NLS case, we explore the relative computational efficiency of the above algorithms by setting $n = 1001$, and plotting the trajectory error over the time interval $[0, 1]$ vs. the total computational cost in Fig. 8. We did not implement

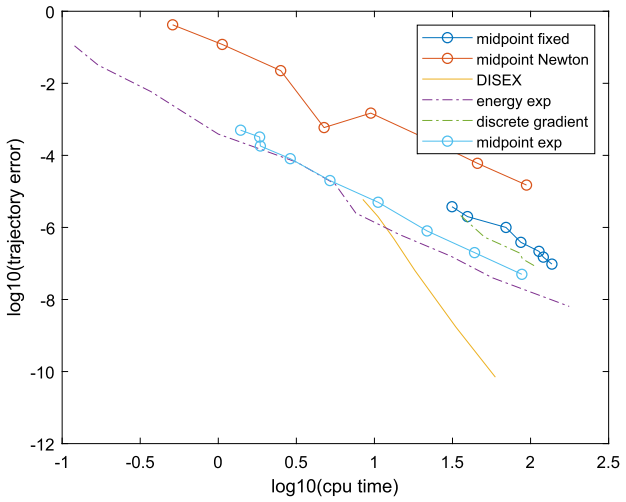
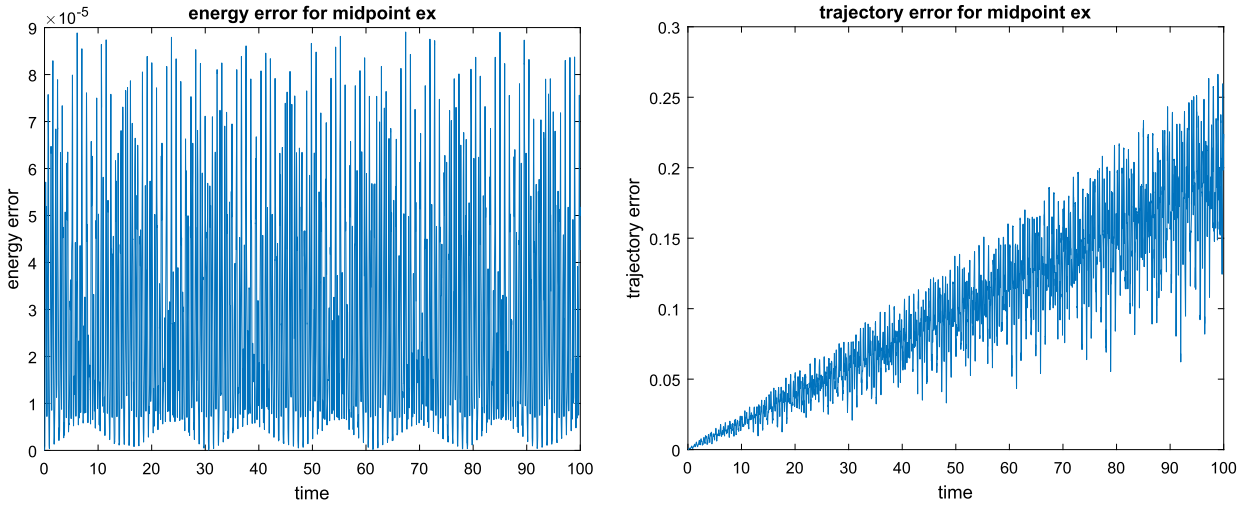


Fig. 5. Comparison of trajectory error vs. CPU time for the nonlinear Schrödinger equation.

Table 4
Maximum timestep and average iteration number for convergence for the KdV equation, as a function of the numerical integrator, nonlinear solver, and spatial resolution.

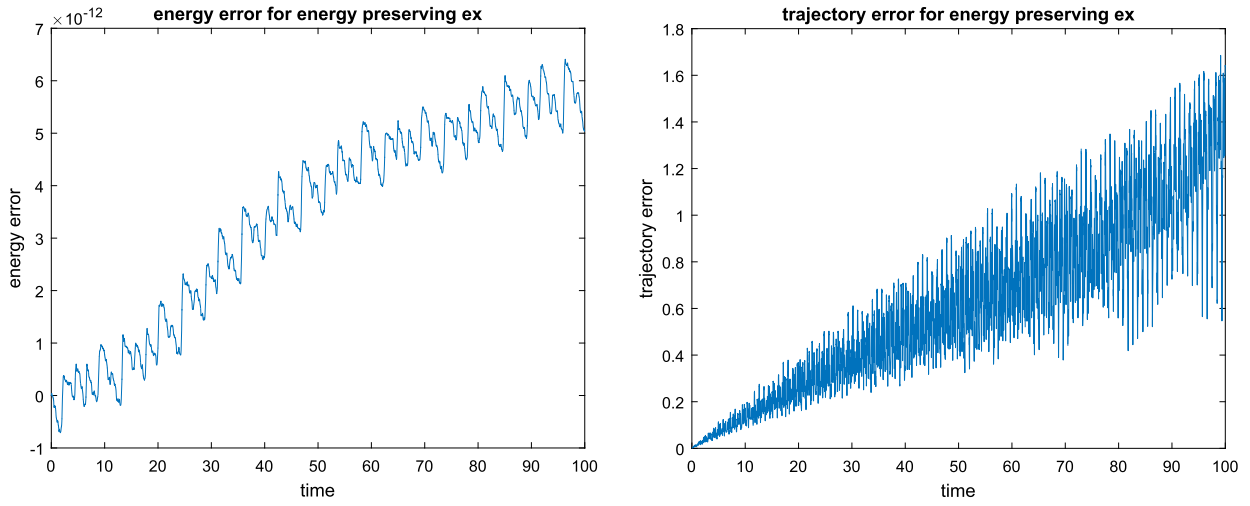
n	Midpoint				Midpoint exp		Discrete gradient		Energy exp		DISEX	
	Fixed point		Newton		Fixed point		Fixed point		Fixed point		Fixed point	
	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$	h_{max}	$iter_{avg}$
401	4×10^{-4}	17.9	4×10^{-4}	25.5	8×10^{-4}	8.4	4×10^{-4}	15.4	0.005	14.9	4×10^{-4}	26.9
601	1×10^{-4}	4.2	1×10^{-4}	7.3	6×10^{-4}	8.6	1×10^{-4}	3.4	0.005	14.8	4×10^{-4}	48.0
801	6×10^{-5}	18.5	5×10^{-5}	4.4	6×10^{-4}	9.4	5×10^{-5}	3.5	0.005	14.9	1×10^{-4}	19
1001	3×10^{-5}	7.7	3×10^{-5}	11.0	6×10^{-4}	10.6	3×10^{-5}	14.9	0.005	14.8	1×10^{-4}	19
1201	1×10^{-5}	2	1×10^{-5}	2	6×10^{-4}	11.4	1×10^{-5}	1	0.005	14.8	1×10^{-4}	22
1401	1×10^{-5}	1	1×10^{-5}	2	6×10^{-4}	16.1	1×10^{-5}	1	0.005	14.8	1×10^{-4}	34



(A) Energy error (B) Trajectory error

Fig. 6. Error plots for the exponential midpoint rule applied to the KdV equation, $n = 401$, $h = 5\times10^{-4}$.

the Newton based solvers here, as we extrapolated that it would require approximately 4×10^4 seconds to compute the trajectory over the time interval $[0, 1]$, so they are not included in the figure. As can be seen from the graph, the minimum computational cost achievable by the energy exponential method is the lowest of the methods considered, as it is capable of taking the largest timestep stably. In addition, it has the highest accuracy for a given computational cost amongst all the low order methods. The method with the next lowest minimum computational cost is the exponential midpoint rule, followed



(A) Energy error (B) Trajectory error

Fig. 7. Error plots for the energy preserving exponential integrator applied to the KdV equation, $n = 401$, $h = 0.005$.

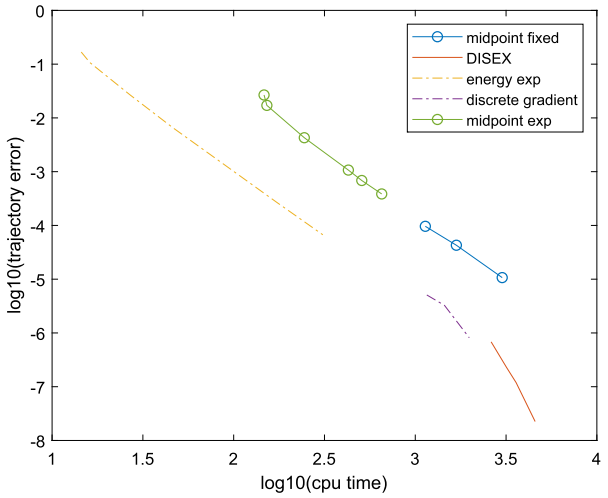


Fig. 8. Comparison of trajectory error vs. CPU time for the KdV equation.

by the midpoint rule with fixed point iteration, and the discrete gradient methods. As before, DISEX still has the steepest slope, which indicates that it is a higher-order method. But, the disadvantage is that the maximum possible timestep is not as large this time, which results in it having the largest minimum computational effort amongst all the methods considered.

6. Summary

For semilinear Poisson system, we have developed two types of geometric exponential integrators, one that preserves the Poisson structure, and the other preserves the energy. They allow fixed point iteration methods to be used for significantly larger timesteps as compared to non-exponential integrators, such as the classical midpoint rule and the standard discrete gradient method, which require the use of Newton type iterations to converge with comparably large timesteps. This results in substantial computational savings, particularly when applied to the simulation of semi-discretized Hamiltonian PDEs. They also exhibit long time stability and conservation of the first integrals. We notice that in practice, energy preserving exponential integrators are more stable and allow for larger timesteps than symplectic exponential integrators. In future work, we will develop higher-order energy preserving exponential integrators.

Acknowledgements

We would like to thank the referees for their helpful comments and suggestions. This research has been supported in part by NSF under grants DMS-1010687, CMMI-1029445, DMS-1065972, CMMI-1334759, DMS-1411792, DMS-1345013, DMS-1813635, and by AFOSR under grant FA9550-18-1-0288.

References

[1] K. Atkinson, W. Han, Theoretical Numerical Analysis: A Functional Analysis Framework, third edition, Texts in Applied Mathematics, vol. 39, Springer, Dordrecht, 2009.
[2] D. Cohen, L. Gauckler, E. Hairer, C. Lubich, Long-term analysis of numerical integrators for oscillatory Hamiltonian systems under minimal non-resonance conditions, BIT Numer. Math. 55 (3) (2015) 705–732.
[3] L. Gauckler, Numerical long-time energy conservation for the nonlinear Schrödinger equation, IMA J. Numer. Anal. 37 (4) (2017) 2067–2090.
[4] E. Hairer, C. Lubich, G. Wanner, Structure-preserving algorithms for ordinary differential equations, in: Geometric Numerical Integration, second edition, in: Springer Series in Computational Mathematics, vol. 31, Springer-Verlag, Berlin, 2006.
[5] M. Hochbruck, A. Ostermann, Exponential Runge–Kutta methods for parabolic problems, Appl. Numer. Math. 53 (2–4) (2005) 323–339.
[6] M. Hochbruck, A. Ostermann, Exponential integrators, Acta Numer. 19 (2010) 209–286.
[7] Z. Kalogiratou, Th. Monovasilis, Diagonally implicit symplectic Runge–Kutta methods with special properties, Appl. Math. Inf. Sci. 9 (1L) (2015) 11–17.
[8] B. Leimkuhler, S. Reich, Simulating Hamiltonian Dynamics, Cambridge Monographs on Applied and Computational Mathematics, vol. 14, Cambridge University Press, Cambridge, 2004.
[9] J.E. Marsden, M. West, Discrete mechanics and variational integrators, Acta Numer. 10 (2001) 357–514.
[10] B. Minchev, Exponential Integrators for Semilinear Problems, PhD thesis, University of Bergen, 2004.
[11] G.R.W. Quispel, G.S. Turner, Discrete gradient methods for solving ODEs numerically while preserving a first integral, J. Phys. A 29 (13) (1996) L341–L349.
[12] J.C. Simo, O. Gonzalez, Assessment of energy-momentum and symplectic schemes for stiff dynamical systems, in: Proc. ASME Winter Annual Meeting, New Orleans, Louisiana, 1993.
[13] L.N. Trefethen, Spectral Methods in MATLAB, Software, Environments, and Tools, vol. 10, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.
[14] X. Wu, K. Liu, W. Shi, An extended discrete gradient formula for multi-frequency oscillatory Hamiltonian systems, in: Structure-Preserving Algorithms for Oscillatory Differential Equations II, Springer, 2015, pp. 95–115.
[15] W.J. Zhu, M.Z. Qin, Poisson schemes for Hamiltonian systems on Poisson manifolds, Comput. Math. Appl. 27 (12) (1994) 7–16.