Fourth-Order Algorithms for Solving the Imaginary Time Gross-Pitaevskii Equation in a Rotating Anisotropic Trap

Siu A. Chin
Department of Physics, Texas A&M University, College Station, TX 77843, USA

Eckhard Krotscheck
Institut für Theoretische Physik, Johannes Kepler Universität Linz, A-4040 Linz, Austria

By implementing the exact density matrix for the rotating anisotropic harmonic trap, we derive a class of very fast and accurate fourth order algorithms for evolving the Gross-Pitaevskii equation in imaginary time. Such fourth order algorithms are possible only with the use of *forward*, positive time step factorization schemes. These fourth order algorithms converge at time-step sizes an order-of-magnitude larger than conventional second order algorithms. Our use of time-dependent factorization schemes provides a systematic way of devising algorithms for solving this type of nonlinear equations.

I. INTRODUCTION

The dynamics of a fast rotating Bose-Einstein condensate (BEC) has been studied extensively in terms of the Gross-Pitaevskii (GP) equation[1,2]. By evolving the GP equation in imaginary time, it is easy to determine the ground state properties of the condensate, such as the formation of vortex-arrays and giant vortices[2,3]. It has been known for some time that the first order pseudo-spectral, split-operator method[4] is a very fast way of solving the non-linear Schrödinger equation. However, first or second order split operator (SO) methods[5,6] and Crank-Nickolson (CN) algorithms with[7] or without[8] splitting ignore the time-dependence of the non-linear potential and converge linearly or quadratically only at very small time steps. Bandaruk and Shen[9] have applied higher order decomposition schemes with negative coefficients to solve the real time non-linear Schrödinger equation. Due to the difficulty of estimating the non-linear potential at intermediate time, they have not demonstrated that their higher order algorithms actually converge with accuracy beyond second order. In any case, their negative coefficient algorithms cannot be used for imaginary time evolution because negative time steps will result in an unbounded diffusion kernel[10,11,12,13].

In this work, we derive a class of very accurate fourth order factorization algorithm for solving the GP equation in imaginary time. These algorithms are made possible by the confluence of three key ideas: 1) The density matrix for a rotating anisotropic harmonic oscillator can be solved exactly. 2) The time-dependence of the non-linear potential can be systematically accounted for in factorization algorithms. 3) Forward, all positive time step algorithms[14,15,16,17,18,19] are now available for solving imaginary time evolution equations.

In the next section, we show how the density matrix of the harmonic oscillator can be exactly implemented as an algorithm. This obviates the need to expand in harmonic eigenstates[6,20]. In Section III, by exact diagonalization, we generalize the result to the case of a rotating anisotropic harmonic trap. In Section IV, we describe the time-dependent form of the fourth-order forward algorithm for solving the GP equation. In Section V, we compare the convergence of various algorithms. We summarize our conclusions in Section VI.

II. EXACT ALGORITHM FOR THE HARMONIC OSCILLATOR

Consider the 1-D harmonic oscillator Hamiltonian operator given by

$$H = T + V = \frac{1}{2}p^2 + \frac{1}{2}\omega^2 x^2. \tag{2.1}$$

Its imaginary time propagator (or density matrix) can be exactly decomposed as

$$\mathrm{e}^{-\tau(T+V)} = \mathrm{e}^{-\tau C_V V}\mathrm{e}^{-\tau C_T T}\mathrm{e}^{-\tau C_V V}, \tag{2.2}$$

where C_V and C_T are functions of τ to be determined. To show this, we simply compare the matrix elements on both sides. For the RHS, we have (ignoring normalization factors)

$$\langle x'|\mathrm{e}^{-\tau C_V V}\mathrm{e}^{-\tau C_T T}\mathrm{e}^{-\tau C_V V}|x\rangle = \mathrm{e}^{-\tau C_V \frac{1}{2}\omega^2 x'^2}\mathrm{e}^{-\frac{1}{2\tau C_T}(x'-x)^2}\mathrm{e}^{-\tau C_V \frac{1}{2}\omega^2 x^2}. \tag{2.3}$$

For the LHS of (2.2), the exact density matrix element is known[21]

$$\begin{aligned}\langle x'|\mathrm{e}^{-\tau(T+V)}|x\rangle &= \exp\biggl(-\frac{\omega}{2\sinh(\omega\tau)}\Bigl[(x'^2+x^2)\cosh(\omega\tau)-2x'x\Bigr]\biggr)\\ &= \exp\biggl(-\frac{\omega}{2\sinh(\omega\tau)}\Bigl[(x'^2+x^2)(\cosh(\omega\tau)-1)+(x'-x)^2\Bigr]\biggr),\end{aligned} \tag{2.4}$$

where we have expressed $-2x'x=(x'-x)^2-x'^2-x^2$. Comparing (2.3) to (2.4) allows us to identify the coefficient functions as

$$C_V=\frac{\cosh(\omega\tau)-1}{\omega\tau\sinh(\omega\tau)} \quad \text{and} \quad C_T=\frac{\sinh(\omega\tau)}{\omega\tau}. \tag{2.5}$$

In the limit of $\tau\to 0$, we have

$$C_V = \frac{1}{2}-\frac{1}{24}\omega^2\tau^2+\frac{1}{240}\omega^4\tau^4+\cdots, \tag{2.6}$$

$$C_T = 1+\frac{1}{6}\omega^2\tau^2+\frac{1}{120}\omega^4\tau^4+\cdots. \tag{2.7}$$

If we keep only the first term, we have a second order algorithm. Keeping the first two terms gives a fourth order algorithm, keeping the first three terms gives a sixth order algorithm, etc..

The exact factorization (2.2) is possible for the harmonic oscillator because its Hamiltonian is quadratic and higher order commutators are either zero or simply proportional to T or V. The harmonic oscillator is characterized by two key commutators,

$$[V,[T,V]]=2\omega^2V, \tag{2.8}$$

$$[T,[V,T]]=2\omega^2T. \tag{2.9}$$

Because of these two equalities, all higher order commutators can be subsumed back to the original operators T and V. To see how this exact decomposition comes about, let's begin with the simple second order decomposition,

$$\begin{aligned}\mathrm{e}^{-\frac{1}{2}\tau V}\mathrm{e}^{-\tau T}\mathrm{e}^{-\frac{1}{2}\tau V} &= \exp\Bigl[-\tau(T+V)+\frac{1}{24}\tau^3([V,[T,V]]-2[T,[V,T]])+O(\tau^5)\Bigr],\\ &= \exp\Bigl[-\tau(T+V)+\frac{1}{24}\tau^3(2\omega^2V-4\omega^2T)+O(\tau^5)\Bigr].\end{aligned} \tag{2.10}$$

Since the error terms are proportional to the original operators, they can be symmetrically moved back to the LHS to yield,

$$\mathrm{e}^{-\tau(\frac{1}{2}-\frac{1}{24}\omega^2\tau^2)V}\mathrm{e}^{-\tau(1+\frac{1}{6}\omega^2\tau^2)T}\mathrm{e}^{-\tau(\frac{1}{2}-\frac{1}{24}\omega^2\tau^2)V}=\mathrm{e}^{-\tau(T+V)+O(\tau^5)}. \tag{2.11}$$

The decomposition of the LHS is then correct to fourth order. The coefficients agree with the expansion (2.6) and (2.7). This example makes it clear that the exact expansion only depends on the abstract commutator relations (2.8) and (2.9), and is independent of the specific representation of the 1-D harmonic oscillator. Also, if we exchange the operators $T\leftrightarrow V$, the coefficients are unchanged. Thus we can also factorize exactly via

$$\mathrm{e}^{-\tau(T+V)}=\mathrm{e}^{-\tau C_VT}\mathrm{e}^{-\tau C_TV}\mathrm{e}^{-\tau C_VT}. \tag{2.12}$$

For real time propagation, we only need to set $\tau=it$ to get the corresponding coefficients,

$$C_V=\frac{1-\cos(\omega t)}{\omega t\sin(\omega t)} \quad \text{and} \quad C_T=\frac{\sin(\omega t)}{\omega t}. \tag{2.13}$$

For either real or imaginary time evolution, one iterates the discretized wave function forward in time via

$$|\psi(\tau+\Delta\tau)\rangle=\mathrm{e}^{-\Delta\tau(T+V)}|\psi(\tau)\rangle \tag{2.14}$$

If the exact density matrix (2.4) were used directly in coordinate space, that would incur a slow, $N\times N$ matrix multiplication of the Gaussian kernel $\mathrm{e}^{-\frac{1}{2\tau C_T}(x'-x)^2}$. The advantage of the factorized form is that this matrix multiplication can be avoided by going to k-space via FFT and multiplying the k-space wave function point-by-point by $\mathrm{e}^{-\tau C_T\frac{1}{2}k^2}$. This is then an order $N\ln_2 N$ operation, much faster than the $N\times N$ coordinate space matrix multiplication.

III. EXACT ALGORITHM FOR A ROTATING ANISOTROPIC HARMONIC TRAP

Consider now the case of an rotating anisotropic harmonic potential with Hamiltonian

$$H = \frac{1}{2}(p_x^2 + p_y^2) + \frac{1}{2}\tilde{\omega}_x^2 x^2 + \frac{1}{2}\tilde{\omega}_y^2 y^2 - \tilde{\Omega}\,(xp_y - yp_x)\,. \tag{3.1}$$

This is a well-studied problem in nuclear physics[22]. Its diagonalization is greatly simplified[23] if we characterize the anisotropy via the deformation parameter δ

$$\tilde{\omega}_x^2 = (1+\delta)\omega_0^2, \quad \tilde{\omega}_y^2 = (1-\delta)\omega_0^2, \tag{3.2}$$

measure lengths in units of the oscillator length $l = 1/\sqrt{\omega_0}$, and express H and $\tilde{\Omega}$ in units of ω_0. The resulting dimensionless Hamiltonian is then

$$H = \frac{1}{2}(p_x^2 + p_y^2) + \frac{1}{2}(1+\delta)x^2 + \frac{1}{2}(1-\delta)y^2 - \Omega\,(xp_y - yp_x)\,, \tag{3.3}$$

where $\Omega = \tilde{\Omega}/\omega_0$. To diagonalize this Hamiltonian, we introduce two new sets of canonical variables,

$$Q_1 = \alpha_1(cx - sp_y), \quad P_1 = \frac{1}{\alpha_1}(cp_x + sy), \tag{3.4}$$

$$Q_2 = \alpha_2(cy - sp_x), \quad P_2 = \frac{1}{\alpha_2}(cp_y + sx), \tag{3.5}$$

where α_i are normalization constants, and $c = \cos(\phi)$, $s = \sin(\phi)$. One can check that the canonical commutator relations are indeed satisfied,

$$[Q_i, P_j] = \mathrm{i}\delta_{ij}\,. \tag{3.6}$$

In terms of $\{Q_i, P_i\}$, because of the way we have parameterized the anisotropy and expressed everything in terms of ω_0, the coefficients of *both* P_2Q_1 and P_1Q_2 can be made to vanish with a single condition:

$$\tan(2\phi) = \frac{2\Omega}{\delta}. \tag{3.7}$$

Using α_i to normalize the P_i^2 terms with unit coefficient, the resulting Hamiltonian can be written as

$$H = T_1 + V_1 + T_2 + V_2 = \frac{1}{2}P_1^2 + \frac{1}{2}\Omega_1^2 Q_1^2 + \frac{1}{2}P_2^2 + \frac{1}{2}\Omega_2^2 Q_2^2, \tag{3.8}$$

where

$$\begin{aligned} \alpha_1^{-2} &= 1 - \frac{\delta}{2} + \frac{1}{2}\sqrt{\delta^2 + 4\Omega^2}\,, \\ \alpha_2^{-2} &= 1 + \frac{\delta}{2} - \frac{1}{2}\sqrt{\delta^2 + 4\Omega^2}\,, \end{aligned} \tag{3.9}$$

with

$$\begin{aligned} \Omega_1^2 &= 1 + \Omega^2 + \sqrt{\delta^2 + 4\Omega^2}\,, \\ \Omega_2^2 &= 1 + \Omega^2 - \sqrt{\delta^2 + 4\Omega^2}\,. \end{aligned} \tag{3.10}$$

Also, from (3.7), we have

$$\begin{aligned} 2s^2 &= 1 - \frac{\delta}{\sqrt{\delta^2 + 4\Omega^2}}\,, \\ 2c^2 &= 1 + \frac{\delta}{\sqrt{\delta^2 + 4\Omega^2}}\,. \end{aligned} \tag{3.11}$$

At $\Omega = 0$, the phase angle $\phi = 0$. As Ω increases, the phase angle approaches 45° asymptotically. Thus s and c in (3.11) are both positive. However, as Ω increases, Ω_2^2 crosses zero and becomes negative at $\Omega = \sqrt{1-\delta}$. At this critical rotation rate, the Coriolis force overcomes the weaker harmonic potential in the y-direction and the anisotropic harmonic oscillator is unstable. Ω_2^2 emerges positive again when α_2^{-2} crosses zero and turn negative at $\Omega^2 = 1+\delta$. Thus Ω_2^2 is negative over the interval $1-\delta \le \Omega^2 \le 1+\delta$. This is an instability of the rotating harmonic oscillator, not necessary that of the Gross-Pitaevskii equation. We will come back to this point in Section VI. Note also that for $\delta = 0$, the algorithm is stable up to $\Omega = 1$.

Eq. (3.8) consists of two independent harmonic oscillator with different frequency. The two exact algorithms must be applied in sequence. However, since T_1 and V_2 only depend on p_x and y, they should be placed next to each other so that both can be evaluated in the same mixed representation described below. Similarly, T_2 and V_1 only depend on x and p_y. We therefore use the following factorization for each algorithm,

$$\mathrm{e}^{-\tau(T_1+V_1)} = \mathrm{e}^{-\tau C_V(1)T_1}\mathrm{e}^{-\tau C_T(1)V_1}\mathrm{e}^{-\tau C_V(1)T_1}, \tag{3.12}$$

$$\mathrm{e}^{-\tau(T_2+V_2)} = \mathrm{e}^{-\tau C_V(2)V_2}\mathrm{e}^{-\tau C_T(2)T_2}\mathrm{e}^{-\tau C_V(2)V_2}, \tag{3.13}$$

and interlaced them as follow:

$$\mathrm{e}^{-\tau(T_1+V_1+T_2+V_2)} = \mathrm{e}^{-\tau C_V(1)T_1-\tau C_V(2)V_2}\mathrm{e}^{-\tau C_T(1)V_1-\tau C_T(2)T_2}\mathrm{e}^{-\tau C_V(1)T_1-\tau C_V(2)V_2}. \tag{3.14}$$

Here we use the shorthand notations $C_V(1) = C_V(\Omega_1)$, $C_T(2) = C_T(\Omega_2)$, etc.. To implement (3.14), let

$$\psi(x,y) = \frac{1}{\sqrt{2\pi}}\int dp_x\, \psi(p_x,y)\,\mathrm{e}^{\mathrm{i}p_x x}, \tag{3.15}$$

$$\psi(p_x,y) = \frac{1}{\sqrt{2\pi}}\int dx\, \psi(x,y)\,\mathrm{e}^{-\mathrm{i}p_x x}, \tag{3.16}$$

and

$$\begin{aligned}\psi(x,p_y) &= \frac{1}{\sqrt{2\pi}}\int dy\, \psi(x,y)\,\mathrm{e}^{-\mathrm{i}p_y y},\\ &= \frac{1}{2\pi}\int dy\, dp_x\, \psi(p_x,y)\,\mathrm{e}^{\mathrm{i}p_x x-\mathrm{i}p_y y}.\end{aligned} \tag{3.17}$$

The operators T_1 and V_2 are diagonal in the representation $\psi(p_x,y)$ and T_2 and V_1 are diagonal in the representation $\psi(x,p_y)$. In practice, $\psi(x,y)$ is discretized as an $N\times N$ complex array and its Fourier transform is computed using the discretized FFT. Thus the exact algorithm consists of four steps:

1. Compute the forward N-1D transform $\psi(p_x,y)$ from $\psi(x,y)$ and multiply $\psi(p_x,y)$ grid-point by grid-point by $\mathrm{e}^{-\tau C_V(1)T_1-\tau C_V(2)V_2}$, where T_1 and V_2 are now understood to be functions of p_x and y.
2. Compute the 2D transform $\psi(x,p_y)$ from the updated $\psi(p_x,y)$ and multiply $\psi(x,p_y)$ by $\mathrm{e}^{-\tau C_T(1)V_1-\tau C_T(2)T_2}$, where V_1 and T_2 are now functions of x and p_y.
3. Compute the inverse 2D transform from the updated $\psi(x,p_y)$ back to $\psi(p_x,y)$ and multiply $\psi(p_x,y)$ by $\mathrm{e}^{-\tau C_V(1)T_1-\tau C_V(2)V_2}$.
4. Compute the backward N-1D transform from the updated $\psi(p_x,y)$ back to $\psi(x,y)$.

Thus the algorithm can be implemented with only three 2D-FFT. (One 2D-transform = $2N$ 1D-transforms.) This is only one 2D-FFT more than solving the non-rotational case.

IV. SOLVING THE GROSS-PITAEVSKII EQUATION

Denoting now the entire rotating trap Hamiltonian (3.8) as the operator

$$T = T_1 + V_1 + T_2 + V_2, \tag{4.1}$$

the corresponding 2D Gross-Pitaevskii equation is

$$(T + g|\psi|^2)\psi(x,y) = \mu\psi(x,y). \tag{4.2}$$

The condensate ground state can be projected out by imaginary time evolution:

$$\psi_0 \propto \lim_{\tau\to\infty} \psi(\tau) = \lim_{\tau\to\infty} \mathrm{e}^{-\tau[T+V(\tau)]+\tau\mu}\psi(0). \tag{4.3}$$

The chemical potential μ is determined by preserving the wave function's normalization to unity. This will be taken for granted and this term will be ignore in the following discussion. Since $\psi(\tau)$ is time-dependent, we have explicitly indicated that the Gross-Pitaevskii potential

$$V(\tau) = g|\psi(\tau)|^2, \tag{4.4}$$

is also time-dependent.

In general, to solve (4.3) by factorization algorithms, one must apply rules of time-dependent factorization[16,24]: *the time-dependent potential must be evaluated at an intermediate time equal to the sum of time steps of all the T operators to its right.* For example, the first order algorithm 1A is

$$\psi(\Delta\tau) = \mathrm{e}^{-\Delta\tau T}\mathrm{e}^{-\Delta\tau V(0)}\psi(0) \tag{4.5}$$

and the first order algorithm 1B is

$$\psi(\Delta\tau) = \mathrm{e}^{-\Delta\tau V(\Delta\tau)}\,\mathrm{e}^{-\Delta\tau T}\psi(0). \tag{4.6}$$

While algorithm 1A is straightforward, 1B requires that the potential be determined from the wave function to be computed. This self-consistency condition can be solved by iterative methods described below.

In contrast to real time propagation, the wave function in imaginary time converges quickly to an *approximate* ground state depending on $\Delta\tau$ and produces a $\psi(\Delta\tau)$ that differs from $\psi(0)$ only by a normalization constant. *Thus after some initial iterations, the normalized $g|\psi(\Delta\tau)|^2$ is independent of τ and can be replaced by $g|\psi(0)|^2$.* This replacement can be justified only at small $\Delta\tau$ when the approximate wave function is close to the exact ground state and is unchanging in time. (For real time propagation, the wave function is always changing with time, and one cannot justify this replacement even at small Δt.) At larger $\Delta\tau$, the approximate ground state may not be a discrete bound state and the algorithm may fail catastrophically. Thus if one approximates $g|\psi(\Delta\tau)|^2$ by $g|\psi(0)|^2$ in (4.6), then the algorithm is still first order, but only at very small $\Delta\tau$. We shall refer to this version of the algorithm as 1B0.

We define the second order algorithm 2A as

$$\psi(\Delta\tau) = \mathrm{e}^{-\frac{1}{2}\Delta\tau V(\Delta\tau)}\mathrm{e}^{-\Delta\tau T}\mathrm{e}^{-\frac{1}{2}\Delta\tau V(0)}\psi(0) \tag{4.7}$$

and algorithm 2B as

$$\psi(\Delta\tau) = \mathrm{e}^{-\frac{1}{2}\Delta\tau T}\mathrm{e}^{-\Delta\tau V(\Delta\tau/2)}\mathrm{e}^{-\frac{1}{2}\Delta\tau T}\psi(0). \tag{4.8}$$

Similarly, one can replace $g|\psi(\Delta\tau)|^2$ by $g|\psi(0)|^2$ in algorithm 2A without affecting its quadratic convergence at very small $\Delta\tau$. We shall refer to this version of the algorithm as 2A0. Algorithm 2B requires two executions of the exact algorithm (3.14) for similar convergence, which is less efficient. We therefore did not implement algorithm 2B.

Fig. 1 shows the convergence of algorithm 1A and 1B0 for the chemical potential μ. Both are very linear at small $\Delta\tau$. The calculation is done for $\delta = 0.5$, $\Omega = 0.5$, and $g = 50$. This choice corresponds to sizable anisotropy, rotation, coupling strength and not close to any particular limit. The calculation uses 64^2 grid points over a 14^2 harmonic length square centered on the origin. Changing the grid size to 128^2 only changes the stable results in the fifth or sixth decimal place. The ground state wave function is nearly converged by $\tau = 2$. The chemical potential shown is calculated at $\tau = 10$. Note that the linear convergence line for 1B0 fails abruptly at $\Delta\tau \approx 0.15$. Since algorithm 2A0 is just running algorithm 1A first followed by 1B0 at half the time-step size, the convergence failure of 1B0 accounts for the failure of algorithm 2A0 near $\Delta\tau \approx 0.3$. Both algorithm 1A and 1B0 require an exceedingly small $\Delta\tau$ (< 0.001) to produce an accurate value of μ. Even algorithm 2A0 requires $\Delta\tau$ to be ≈ 0.05.

V. SELF-CONSISTENT ITERATIONS

To see the full effect of time-dependent factorization, we must implement 1B and 2A in the form (4.6) and (4.7) with self-consistent iterations to determined the GP potential. The required consistency equation is of the form

$$\psi = \frac{1}{\sqrt{Z}} \mathrm{e}^{-\frac{1}{2}b|\psi|^2}\phi, \tag{5.1}$$

where $b = c\Delta\tau g$ for some coefficient c, ϕ is the unnormalized intermediate wave function prior to the evaluation of the potential term, and Z is the constant that normalizes ψ. This is needed because we are solving for the GP potential, which requires a normalized wave function. Since only the square of the modulus is needed, we solve (5.1) as

$$x = \frac{1}{Z}\mathrm{e}^{-bx}a \tag{5.2}$$

where $x = |\psi(\mathbf{r})|^2$, $a = |\phi(\mathbf{r})|^2$ and

$$Z = \int \mathrm{e}^{-bx(\mathbf{r})}a(\mathbf{r})\, d\mathbf{r} \approx \sum_i \mathrm{e}^{-bx_i}a_i\,(\Delta x)^2. \tag{5.3}$$

It is helpful to view these equations in the discrete forms in which they are actually solved. When necessary, we will denote array elements explicitly as $x_i = |\psi(\mathbf{r}_i)|^2$, etc..

A naive way of solving (5.2) is just to iterate,

$$x_{n+1} = F(x_n) = \frac{1}{Z(x_n)}\mathrm{e}^{-bx_n}a. \tag{5.4}$$

Starting with $x_0 = 0$, the first iteration would produces the normalized $\tilde{a} = a/Z$, which is a reasonable starting guess. Denoting $x_n = x^* + \varepsilon_n$, where x^* is the exact solution, then

$$\begin{aligned} \varepsilon_{n+1} &= F'(x^*)\,\varepsilon_n + O(\varepsilon_n^2) \\ &= -bx^*\,(1 - O(1/N))\,\varepsilon_n + \cdots \\ &\approx -(c\Delta\tau g x^*)\varepsilon_n. \end{aligned} \tag{5.5}$$

The $O(1/N)$ term neglected above is from differentiating $1/Z$ with respect to x_i $(= \mathrm{e}^{-b\,(x_n)_i}a_i/\sum_{j=1}^{N}\mathrm{e}^{-b\,(x_n)_j}a_j)$, where N is the total number of array elements. It is therefore an excellent approximation to regard Z as a constant when differentiating with respect to individual array elements. The error propagation (5.5) explains why this naive iteration is undesirable; it will diverge abruptly at large $\Delta\tau$ such that $|c\Delta\tau g x^*| > 1$.

The self-consistency equation can be solved by better methods, such as Newton's or Halley's iterations. However, at large $\Delta\tau$, the normalized $\tilde{a}$ is not a good enough starting point. Fortunately, (5.2) has the exact solution

$$x = \frac{1}{b}W\left(\frac{b\,a}{Z}\right) \tag{5.6}$$

where $W(y)$ is the Lambert W-function[25]

$$W\mathrm{e}^W = y \tag{5.7}$$

with series expansion

$$W(y) = y - y^2 + \frac{3}{2}y^3 - \frac{8}{3}y^4 + O(y^5). \tag{5.8}$$

The series expansion (5.8) is not useful for our purpose since its radius of convergence is only $1/e$. A better choice is the following uniform approximation by Winitzki[26],

$$W(y) \approx \ln(1+y)\left(1 - \frac{\ln[1+\ln(1+y)]}{2+\ln(1+x)}\right), \tag{5.9}$$

which has a maximum relative error of 2% (at $x \approx 2$) in [0,∞].

The normalization constant Z can in principle be determined from

$$b = \int W\left(\frac{b\,a}{Z}\right) d\mathbf{r}, \tag{5.10}$$

but this integral equation is time consuming to solve when $W(y)$ itself has to be computed numerically. A more workable scheme is compute compute Z via (5.3) from x_n, and compute x_{n+1} via (5.6). The use of (5.6) will guarantee convergence at all b and $\Delta\tau$, provided that one can compute $W(y)$ in a simple way. Thus we use the normalized $\tilde{a}$ as the starting value, compute Z via (5.3), solve for x via (5.6) and normalize it to obtain an approximate GP potential at all $\Delta\tau$. We find that this one iteration is sufficient to remove all instability in algorithms 1A and 1B; further Newton-Raphson iterations produce marginal improvments not worth the additional effort.

Fig.1 shows the convergence of algorithms 1B and 2A when the self-consistency condition is approximately satisfied by our one W-function iteration. The results are denoted as 1BW and 2AW. The instability in 1B0 and 2A0 no longer appears. The convergence of both are linear and quadratic out to large values of $\Delta\tau$.

For first and second order algorithms, self-consistent iterations are not needed because $\Delta\tau$ has to be very small in order for these algorithms to produce results close to the exact one. If $\Delta\tau$ is small, then one may as well use 1B0 and 2A0 without wasting time on self-consistent iterations. Self-consistency is a concern only when one is interested in enlarging the step-size convergence of higher order algorithms.

VI. FORWARD FOURTH-ORDER ALGORITHMS

It is well known from studies of symplectic integrators that factorizations of the form (4.5) to (4.8) can be generalized to higher order in the form[27,28,29,30,31,32,33,34]

$$\mathrm{e}^{-\Delta\tau(T+V)} = \prod_i \mathrm{e}^{-a_i\Delta\tau T}\mathrm{e}^{-b_i\Delta\tau V}, \tag{6.1}$$

with coefficients $\{a_i, b_i\}$ determined by the required order of accuracy. However, as first proved by Sheng[35] and made explicit by Suzuki[36], beyond second order, any factorization of the form (6.1) *must* contain some negative coefficients in the set $\{a_i, b_i\}$. Goldman and Kaper[37] later proved that any factorization of the form (6.1) must contain at least one negative coefficient for *both* operators. Since a negative time step for the kinetic energy operator will result in an unbound and unnormalizable wave function, no such factorization scheme can be used to evolve the imaginary time Schrödinger equation, including the Gross-Pitaevskii equation. To go beyond second order, one must use *forward* factorization schemes with only positive factorization coefficients[14,15,16]. These forward algorithms are currently the only fourth order algorithms possible for solving time-irreversible equations with a diffusion kernel[10,11] and have been applied successfully in solving the imaginary time Schrödinger equation[12,13]. Omelyand, Mryglod and Folk[18,19] have compiled an extensive list of fourth and higher order symplectic algorithms. However, their sixth- and eight-order algorithms contain negative time steps and are not forward algorithms. Recently, one of us have proved[38] that while sixth order forward algorithms are possible, they require an additional commutator currently not implementable. Thus forward algorithms are very unique. Here, we will show that they also yield highly accurate fourth order algorithms for solving the Gross-Pitaevskii equation.

The problem we seek to solve is the ground state of

$$H = H_x + H_y + V(x, y, \tau) \tag{6.2}$$

where $V(x, y, \tau)$ is the GP potential (4.4) and

$$H_x = \frac{1}{2}p_y^2 + \frac{1}{2}(1+\delta)x^2 - \Omega x p_y\,, \tag{6.3}$$

$$H_y = \frac{1}{2}p_x^2 + \frac{1}{2}(1-\delta)y^2 + \Omega y p_x\,. \tag{6.4}$$

The Hamiltonian fundamentally has three operators, which are diagonal in (x, y), (x, p_y) and (p_x, y). If the external trapping potential $V_{ext}(x, y)$ is more general and non-harmonic, we can still write,

$$H = H_x + H_y + V(x, y, \tau) \tag{6.5}$$

but now with

$$V(x, y, \tau) = V_{ext}(x, y) - \frac{1}{2}(1+\delta)x^2 - \frac{1}{2}(1-\delta)y^2 + g|\psi(x, y, \tau)|^2. \tag{6.6}$$

The parameter δ is then a free parameter associated with algorithm, which we can choose to match the asymmetry of V_{ext}, or just set to zero. The crucial points is that, for a rotating trap, harmonic or not, the Hamiltonian has three operators diagonal in three separate spaces. By computing the density matrix of

$$T = H_x + H_y \tag{6.7}$$

exactly via algorithm (3.14), we have reduced the Hamiltonian to a two-operator problem. This is a tremendous simplification. This simplification is not restricted to harmonic traps, but holds equally for an arbitrary external potential. The key point is that the *rotating* part of the Hamiltonian can be diagonalized regardless of the choice of the confining potential. When we diagonalize $H_x + H_y$, we *generate* an inverted harmonic potential in (6.6), which must be compensated by the external potential or the GP potential. In the following we will present results only for (6.2), but our algorithm works in the general case of (6.6). We will come back to this point when we discuss over-critical rotation in the next section.

Because implementing T is computational demanding, we must choose a fourth order algorithm with a minimal number of T operators. Thus among the many forward algorithms discovered so far[15,16,17,18,19], we choose to implement only the simplest algorithm, 4A.

$$\psi(\Delta\tau) = \mathrm{e}^{-\frac{1}{6}\Delta\tau V(\Delta\tau)}\mathrm{e}^{-\frac{1}{2}\Delta\tau T}\ \mathrm{e}^{-\frac{2}{3}\Delta\tau \widetilde{V}(\Delta\tau/2)}\ \mathrm{e}^{-\frac{1}{2}\Delta\tau T}\mathrm{e}^{-\frac{1}{6}\Delta\tau V(0)}\psi(0)\ , \tag{6.8}$$

with $\widetilde{V}$ given by

$$\widetilde{V} = V + \frac{\Delta\tau^2}{48}\,[V,[T,V]]\,. \tag{6.9}$$

Despite the seeming complexity of T as defined by the Hamiltonian (3.8), we have remarkably,

$$[V,[T,V]] = \left(\frac{\partial V}{\partial x}\right)^2 + \left(\frac{\partial V}{\partial y}\right)^2. \tag{6.10}$$

Thus the midpoint effective potential is

$$\widetilde{V}(\Delta\tau/2) = g|\psi(\Delta\tau/2)|^2 + \frac{\Delta\tau^2 g^2}{48}\left[\Big(\frac{\partial\,|\psi(\Delta\tau/2)|^2}{\partial x}\Big)^2 + \Big(\frac{\partial\,|\psi(\Delta\tau/2)|^2}{\partial y}\Big)^2\right]. \tag{6.11}$$

(For the more general case, V is given by (6.6)) The partial derivatives can be computed numerically[12] by use of finite differences or FFT. Since the FFT derivative converges exponentially with grid size, the use of FFT derivative is preferable when the system can be made periodic. In the case with bound state wave functions, this can be done by extending the grid size so the the wave function is essentially zero near the grid edge.

To implement this fourth order algorithm, we first replace $\psi(\Delta\tau/2)$ and $\psi(\Delta\tau)$ by $\psi(0)$. We will refer to this as algorithm 4A00. Its convergence is shown in Fig. 2. We have retained some first and second order results for comparison. Aside from its abrupt instability at $\Delta\tau \approx 0.3$, its convergence is remarkably flat. All the results at $\Delta\tau < 0.3$ differ only in the fifth decimal place.

We can also improve the convergence by making the final wave function $\psi(\Delta\tau)$ consistent with $V(\Delta\tau)$. The results for iterating the W-function once as described previously is denoted as 4A0W. By just iterating the W-function once, we extended the convergence out to $\Delta\tau \approx 0.5$. Algorithm 4A00 and 4A0W can achieve the result of 2A0 at $\Delta\tau$ nearly 30 to 40 times times as large.

To fully implement the time-dependent factorization scheme (6.8) and to remove the instability in 4A0W, we evolve the midpoint wave function $\psi(\Delta\tau/2)$ from $\psi(0)$ by a second order algorithm 2AW and iterate the final wave function $\psi(\Delta\tau)$ for consistency. We denote this algorithm as 4AWW. Its convergence is now smooth, stable and fourth order as shown in Fig.2.

As pointed out in Ref. 12, when the eigenfunction converges as

$$\psi = \psi_0 + O(\Delta\tau^n), \tag{6.12}$$

the eigenvalue converges as

$$E = E_0 + O(\Delta\tau^{2n}). \tag{6.13}$$

In Fig. 3, we compare the convergence of the GP ground state energy

$$E = \int \psi^*(\,T + \frac{1}{2}g|\psi|^2\,)\psi d^2\mathbf{r}/\int |\psi|^2 d^2\mathbf{r}. \tag{6.14}$$

All the fitted lines are of the monomial form

$$E - E_0 = C\Delta\tau^n. \tag{6.15}$$

Algorithms 1A and 1B0 yielded near-identical quadratic convergence. Algorithm 2AW can be fitted with a fourth order monomial as shown. The fit is not perfect because our W-function is only an approximation. Algorithm 4A00 and 4A0W failed too abruptly to show a smooth trend, but 4AWW can indeed be fitted with an eighth order monomial.

The computational effort required by each algorithm is essentially that of evaluating the exact algorithm (3.14), which uses 3 2D-FFT. Since 1A, 1B0, and 2A0 all use the exact algorithm once, the second order algorithm 2A0 is clearly superior. Algorithms 4A00 requires two evaluations of the exact algorithm plus the gradient potential. The gradient potential, if done by FFT, requires 2 2D-FFT. Thus algorithm 4A0n requires 8 2D-FFT, which is $8/3 \approx 3$ times the effort of algorithm 2A0. Since algorithm 4A00 converges much better than 2A0 at time-steps more than three times as large, the class of 4A00 and 4A0W algorithm is clearly more efficient. This efficiency is especially evident if higher accuracy is required. The fully implemented algorithms 4AWW use the second order algorithm to evaluate midpoint wave function and is therefore ≈ 4 time the effort of 2A0. Looking at Fig. 2, algorithms 4AWW clearly converge better than 2AW (2A0) even at time-steps four times as large. Note that the first and second order algorithms are basically similar, whereas all fourth order algorithm are qualitatively distinct. The second-order algorithm is not an order-of-magnitude better than a first-order algorithm, whereas all fourth-order algorithms are an order-of-magnitudes better than the second-order algorithm.

This advantage of fourth order algorithms is cumulative. For example, one can quickly evolve into the ground state by use of large time steps. As stated earlier, the GP ground state can be obtained at $\tau = 2$. Using algorithm 4A0W at $\Delta\tau = 0.5$, one can get there in four iterations. Algorithm 2A0 would have taken 80 iterations at $\Delta\tau \approx 0.02$.

To see how these comparisons work out in practice, we give below some timing information. Since running time is code and machine dependent, this should be viewed as merely illustrative. The algorithms were programmed in Fortran 90 and ran on a 1.2 GH Pentium machine with 0.5 GB of RAM. The IMSL 2D-FFT is used. For the case of a 64×64 point mesh. each run of algorithms 1A, 1B0, 2A0, 1BW, and 2AW required 0.0130, 0.0135, 0.0138, 0.0173 and 0.0203 second, respectively. (Timing is obtained by averaging over 500 run of each algorithm.) Algorithms 1A, 1B0, 2A0 are indeed comparable, but each self-consistency iteration increases the time by ≈ 0.006 second, which is an increase of 40% for 2AW. The time for algorithm 4A00, 4A0W and 4AWW are respectively 0.0394, 0.0459, 0.0630 second. The time for algorithm 4A00 is approximately three times that of 2A0 and the self-consistency iteration now accounts for only an 16% increase. Algorithm 4AWW is slightly more than four times that of 2A0. In the case of a 128×128 point mesh, the timing for 2A0, 4A00, 4A0W, 4AWW are respectively 0.0568, 0.1669, 0.1916 and 0.2617 second. Algorithm 4A00 is accurately three times that of 2A0 and 4AWW is ≈ 1.5 times that of 4A00. Everything scaled up approximately by a factor of four.

By solving the density matrix of the rotation harmonic oscillator (6.7) exactly, we have effected a tremendous simplification and has allowed us to derive very compact fourth-order algorithms with excellent large time-step convergence. They are no more difficult to implement than second order algorithms. If we do not have the exact density matrix, then we would have to approximate each occurrence of $\mathrm{e}^{-\frac{1}{2}\Delta\tau T}$ in (6.8) to fourth order, resulting in a much more complex algorithm.

However, by solving the rotating harmonic oscillator exactly, the current algorithms also inherited its limitations. As alluded to in Section III, the rotating harmonic trap becomes unstable at $\Omega_c = \sqrt{1-\delta}$. Thus if we are to use the exact algorithm (3.14), we must require $\Omega < \Omega_c$. However, it is known[39,40] that for $\delta < 1/5$ and g sufficiently large, the full GP equation can support "over-critical" rotation in the interval $\sqrt{1-\delta} < \Omega < \sqrt{1+\delta}$. For over-critical rotation, one must not prematurely impose the limitation of the rotating harmonic oscillator on the algorithm.

In light of our previous discussion, we now consider the case of

$$V_{ext}(x,y) = \frac{1}{2}(1+\Delta)x^2 + \frac{1}{2}(1-\Delta)y^2, \tag{6.16}$$

and group the Hamiltonian as follow

$$H = H_x + H_y + V(x,y,\tau) \tag{6.17}$$

where H_x and H_y are defined as before in (6.3), (6.4), and

$$V(x,y,\tau) = \frac{1}{2}(\Delta-\delta)x^2 - \frac{1}{2}(\Delta-\delta)y^2 + g|\psi(x,y,\tau)|^2. \tag{6.18}$$

We thus divorce the deformation parameter δ associated with the algorithm, from the physical deformation parameter Δ associated with the trapping potential. If we choose $\delta = 0$, the algorithm is stable up to $\Omega = 1$, above the physical critical value of $\Omega_c = \sqrt{1-\Delta}$. Moreover, since in the Thomas-Fermi approximation the density profile follows the shape of the potential, (6.18) indeed suggests that the inverted harmonic potential $-\frac{1}{2}\Delta y^2$ can be compensated by the GP potential at sufficiently large g, making over-critical rotation possible.

VII. CONCLUDING SUMMARY

In this work we have derived a number of fourth order algorithms and demonstrate their fourth order convergence in solving the GP equation in a rotating, anisotropic harmonic trap. These fourth order algorithms, based on forward factorization schemes, are the only class of factorization algorithms possible for solving evolution equations with a diffusion kernel. Our use of the time-dependent factorization rule provided a systematic way of solving the *nonlinear* GP equations and can be generalized to solve similar nonlinear equation such as the Hartree-Fock and the Kohn-Sham equation[41,42]. These fourth order algorithms are particularly efficient in solving for the ground state by use of large time steps. In constrast to other algorithms, generalizing these algorithms to 3D is very transparent, one simply replaces 2D-FFT everywhere by 3D-FFT. Our use of the exact algorithm, which diagonalizes the rotating component of the Hamiltonian, is general and can applied to any external trapping potential. This exact algorithm also provided insight for understanding over-critical rotation. Physical results obtained by applying these algorithms will be presented elsewhere.

Acknowledgments

This work was supported in part, by a National Science Foundation grant (to SAC) No. DMS-0310580 and the Austrian Science Fund FWF (to EK) under project P15083-N08).

REFERENCES

[1] F. Dalfovo, S. Giorgini, L. Pitaevskii and S. Stringari, Rev. Mod. Phys. **71**, 463 (1999).
[2] A. Fetter and A. Svidzinsky, J. Phys.: Condens. Matter **13**, R135 (2001).
[3] A. Fetter, B. Jackson and S. Stringari, Phys. Rev. **A71**, 013605 (2005).
[4] T. R. Taha and M. J. Ablowitz, J. Comput. Phys. **55**, 203 (1984).
[5] B. Jackson, J. F. McCann and C. S. Adams, J. Phys. **B 31**, 4489 (1998).
[6] C. M. Dion and E. Cances, Phys. Rev. **E 67**, 046706 (2003).
[7] S. K. Adhikari and P. Muruganandam, J. Phys. **B 35**, 2831 (2002)
[8] P. A. Ruprecht, M. J. Holland, K. Burnett, and M. Edwards, Phys. Rev. **A 51**, 4704 (1995).
[9] A. D. Bandrauk and H. Shen, J. Phys. **A 27**, 7147 (1994).
[10] H. A. Forbert and S. A. Chin, Phys. Rev. **E 63**, 016703 (2001).
[11] H. A. Forbert and S. A. Chin, Phys. Rev. **B 63**, 144518 (2001).
[12] J. Auer, E. Krotscheck, and S. A. Chin, J. Chem. Phys. **115**, 6841 (2001).
[13] O. Ciftja and S. A. Chin, Phys. Rev. **B 68**, 134510 (2003).
[14] M. Suzuki, *Computer Simulation Studies in Condensed Matter Physics VIII*, eds, D. Landau, K. Mon and H. Shuttler (Springler, Berlin, 1996).
[15] S. A. Chin, Phys. Lett. **A226**, 344 (1997).
[16] S. A. Chin and C. R. Chen, J. Chem. Phys. **117**, 1409 (2002).
[17] S. A. Chin, and C. R. Chen, "Forward Symplectic Integrators for Solving Gravitational Few-Body Problems", arXiv, astro-ph/0304223, in press, Cele. Mech. and Dyn. Astron.
[18] I. P. Omelyan, I. M. Mryglod and R. Folk, Phys. Rev. **E66**, 026701 (2002).
[19] I. P. Omelyan, I. M. Mryglod and R. Folk, Comput. Phys. Commun. **151** 272 (2003)
[20] M. Edwards *et al.*, Phys. Rev. **A53**, R1950 (1996).
[21] R. P. Feynman, *Statistical Mechanics -A set of Lectures*, Benjamin Advanced Book, Reading, MA, 1972.
[22] P. Ring and P. Schuck, "The Nuclear Many-Body Problem", P.133, Springer-Verlag, Berlin-NY (1980).
[23] M.Ö. Oktel, Phys. Rev. **A69**, 023618 (2004).
[24] M. Suzuki, Proc. Japan Acad. **69**, Ser. B, 161 (1993).
[25] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey and D.E. Knuth, Adv. Comput. Math **5**, 329 (1996).
[26] S. Winitzki, *Lecture Notes in Computer Science*, Springer-Verlag, **2667**, 780 (2003).
[27] E. Forest and R. D. Ruth, Physica D **43**, 105 (1990).
[28] M. Creutz and A. Gocksch, Phys. Rev. Letts. **63**, 9 (1989).
[29] H. Yoshida, Phys. Lett. **A150**, 262 (1990).
[30] H. Yoshida, Celest. Mech. **56** (1993) 27.
[31] R. I. McLachlan, SIAM J. Sci. Comput. **16**, 151 (1995).
[32] M. Suzuki, Phys. Lett. **A146**, 319 (1990); **165**, 387 (1992).
[33] R. I. McLachlan and G. R. W. Quispel, Acta Numerica, **11**, 241 (2002).
[34] *Geometric Numerical Integration*, by E. Hairer, C. Lubich, and G. Wanner, Springer-Verlag, Berlin-New York, 2002.
[35] Q. Sheng, IMA Journal of numberical anaysis, **9**, 199 (1989).

[36] M. Suzuki, J. Math. Phys. **32**, 400 (1991).
[37] D. Goldman and T. J. Kaper, SIAM J. Numer. Anal., **33**, 349 (1996).
[38] S. A. Chin, Phys. Rev. E **71**, 016703 (2005).
[39] A. Recati, F. Zambelli and S. Stringari, Phys. Rev. Letts. **86**, 377 (2001).
[40] P. Rosenbusch *et al.*, Phys. Rev. Letts. **88**, 250403 (2002).
[41] M. Aichinger, S. A. Chin, and E. Krotscheck, Computer Physics Communications, in press
[42] M. Aichinger, S. A. Chin, E. Krotscheck, and E. Räsänen, Phys. Rev. B. (submitted)

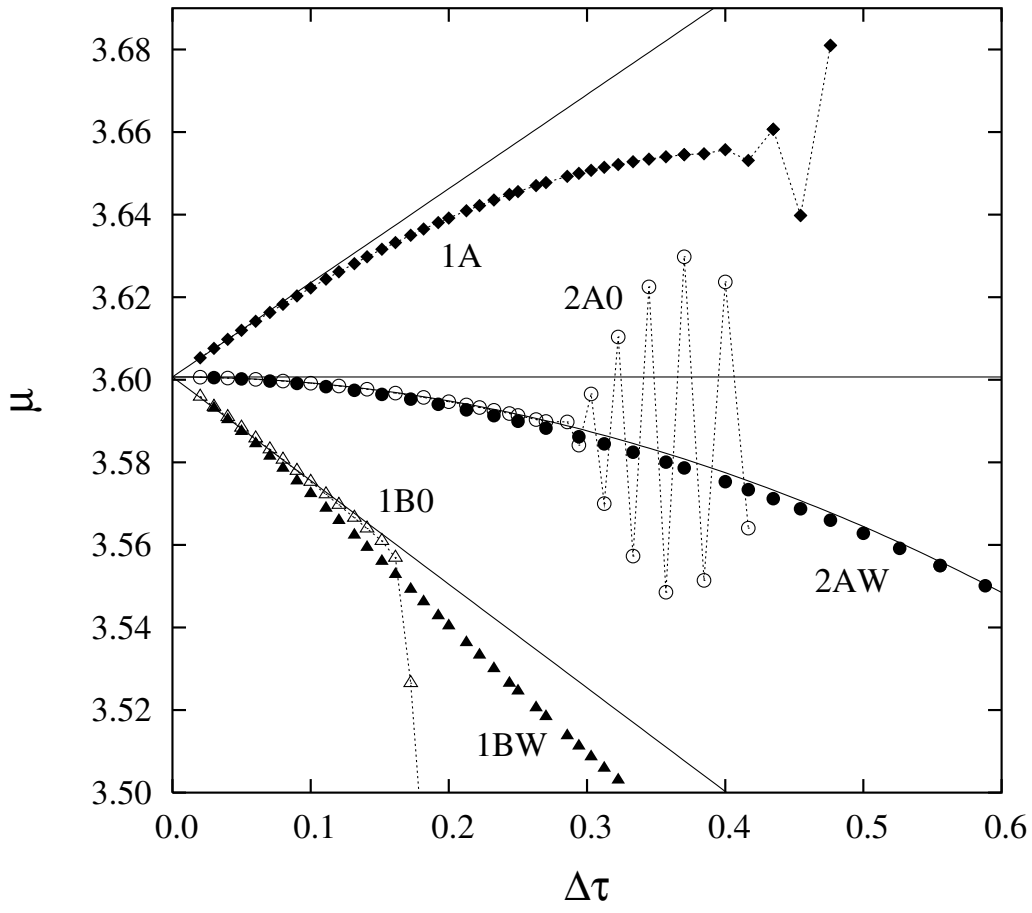

FIG. 1: Comparing the convergence of first and second-order algorithms in computing the chemical potential of the Gross-Pitaevskii equation in a rotating anisotropic trap. The lines are fitted curves to algorithm 1A, 1B0 and 2A0 to demonstrate the order of convergence of each algorithm. The instability of data points in algorithms 1B0 and 2A0 are removed by the inclusion of one self-consistent W-function iteration as indicated by 1BW and 2AW.

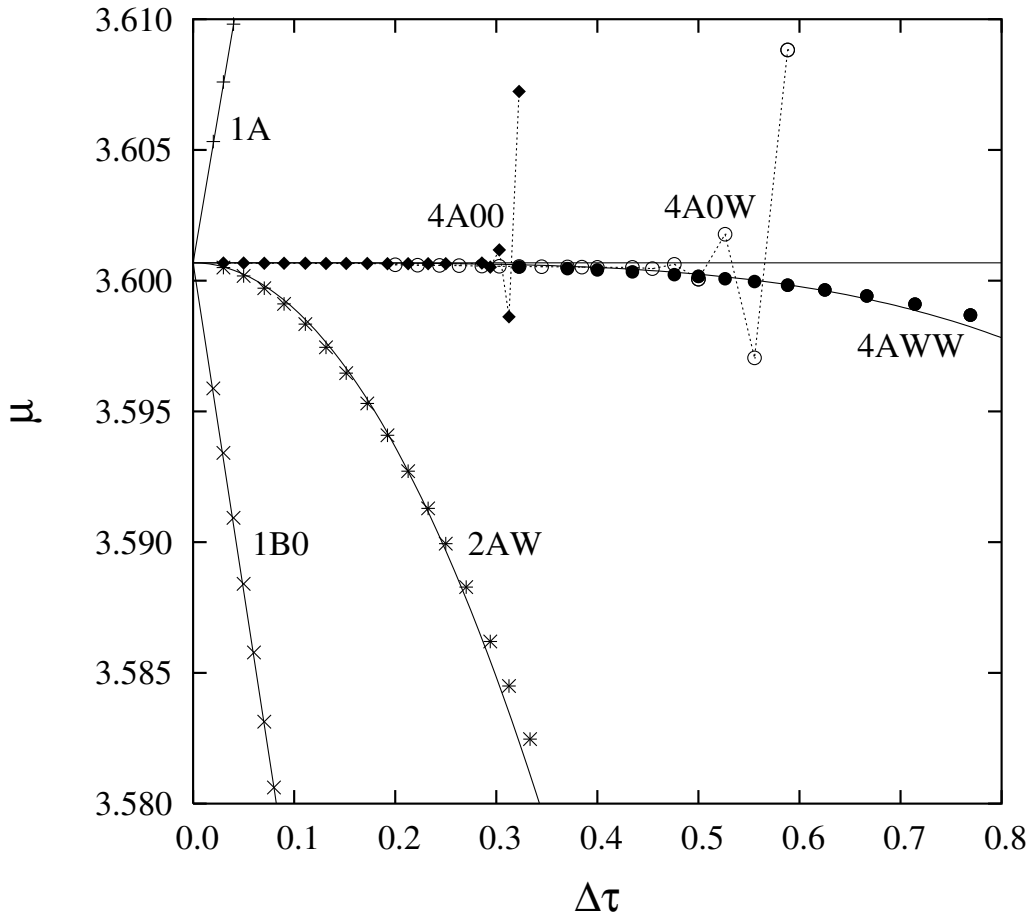

FIG. 2: The convergence of fourth-order algorithms in computing the chemical potential of the Gross-Pitaevskii equation in a rotating trap. Algorithms 4A00 (solid diamonds) and 4A0W (circles) are unstable beyond $\Delta\tau \approx 0.3$ and 0.45 respectively. Algorithm 4AWW (solid circles) is stable and showed excellent fourth order convergence.

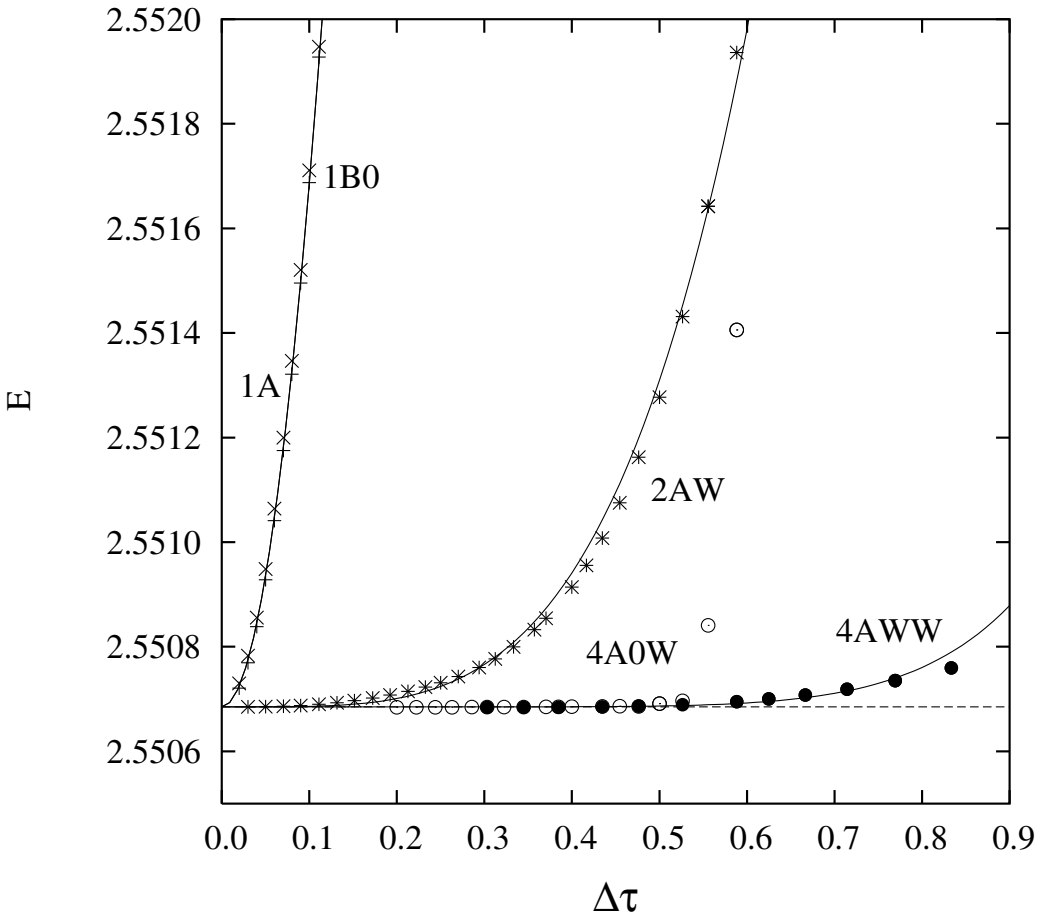

FIG. 3: The convergence of various algorithms in computing the ground state energy of the GP equation. Both first order results showed near-identical quadratic convergence. The second order result 2AW (asterisks) is fourth order and 4AWW (solid circles) is eighth order. Results for 4A0W (circles) cannot be fitted because instability sets in abruptly at $\Delta\tau \approx 0.5$. Results for 4A00 is similar with instability at ≈ 0.3 and is not shown.