Solve the GPE in a 1D parabolic trap

Ashton Bradley

April 12, 2019

1 Introduction

In this simple example we find a ground state of the Gross-Pitaevskii equation in a harmonic trap.

The mean field order parameter evolves according to

$$i\hbar\frac{\partial\psi(x,t)}{\partial t} = \left(-\frac{\hbar^2\partial_x^2}{2m} + V(x,t) + g|\psi(x,t)|^2\right)\psi(x,t)$$

2 Loading the package

First, we load some useful packages.

```
using Plots, LaTeXStrings, Pkg, Revise
gr(legend=false,titlefontsize=12,size=(500,300),transpose=true,colorbar=false);
```

Now load `FourierGPE`

```
using FourierGPE
```

Let's define a convenient plot function

```
function showpsi(x,ψ)
    p1 = plot(x,abs2.(ψ))
    xlabel!(L"x/a_x");ylabel!(L"|\psi|^2")
    p2 = plot(x,angle.(ψ))
    xlabel!(L"x/a_x");ylabel!(L"\textrm{phase}(\psi)")
    p = plot(p1,p2,layout=(2,1),size=(600,400))
    return p
end
```

```
showpsi (generic function with 1 method)
```

3 User parameters

We reserve a place for user parameters.

```
@with_kw mutable struct Params <: UserParams @deftype Float64
    # user parameters:
    κ = 0.1
end
par = Params();
```

Let's set the system size, and number of spatial points

```
L = (20.0,)
N = (128,)
X,K,dX,dK,DX,DK,T = maketransforms(L,N)
espec = 0.5*k2(L...,N...);
```

Now we need to initialize the simulation object and the transforms

```
sim = Sim(L,N,par)
@pack! sim = T,X,K,espec
initsim!(sim)
@unpack_Sim sim;
```

3.1 Declaring the potential

Let's define the trapping potential.

```
import FourierGPE.V
V(x,t) = 0.5*x^2
```

```
V (generic function with 3 methods)
```

We only require that it is a scalar function because alter we will evaluate it using a broadcasted dot-call.

4 Initial condition

Let's define a useful Thomas-Fermi wavefunction

```
ψ0(x,μ,g) = sqrt(μ/g)*sqrt(max(1.0-V(x,0.0)/μ,0.0)+im*0.0)
x = X[1];
```

The initial state is now created as

```
ψi = ψ0.(x,μ,g)
ψi .+= (randn(N...) |> complex)
ϕi = kspace(ψi,sim)
@pack! sim = ϕi;
```

5 Evolution in k-space

The `FFTW` library is used to evolve the Gross-Pitaevskii equation in k-space

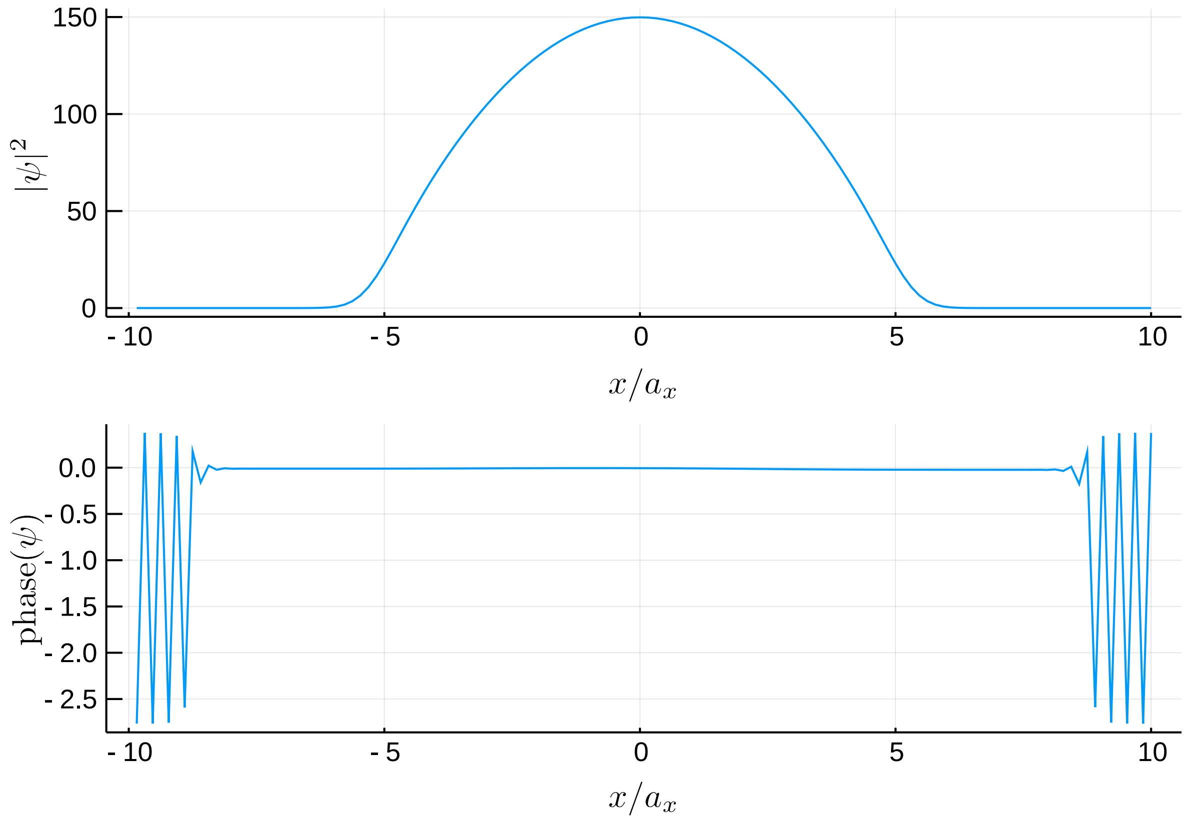
```
sol = runsim(sim.ϕi,sim);
```

```
0.566155 seconds (1.42 M allocations: 68.902 MiB, 5.42% gc time)
```

Here we save the entire solution as a single variable `sol`.

Let's have a look at the final state and verify we have a ground state

```
ϕg = sol[end]
ψg = xspace(ϕg,sim)
showpsi(x,ψg)
```



The initial Thomas-Fermi state has been evolved for a default time $t = 2/\gamma$ which is a characteristic damping time for the dissipative system with dimensionless damping γ. The solution will approch the ground state satisfying $L\psi_0 = \mu\psi_0$ on a timescale of order $1/\gamma$. The figure shows a smooth density profile and a completely homogeneous phase profile over the region of finite atomic density, as required for the ground state.

5.1 Default simulation parameters

The default parameters are given in the declaration of `Sim`, which allows parameter interdependence. The struct `Sim` is declared as:

```
@with_kw mutable struct Sim{D} <: Simulation{D} @deftype Float64
    L::NTuple{D,Float64}
    N::NTuple{D,Int64}
    μ = 15.0
    g = 0.1
    γ = 0.5; @assert γ >= 0.0
    ti = 0.0
    tf = 2/γ
    Nt::Int64 = 200
    t::LinRange{Float64} = LinRange(ti,tf,Nt)
    ϕi::Array{Complex{Float64},D} = zeros(N...) |> complex
    params::UserParams # optional parameters
```

```
    T::TransformLibrary{D} = Transforms{D}()
    X::NTuple{D,Array{Float64,1}} = xvecs(L...,N...)
    K::NTuple{D,Array{Float64,1}} = kvecs(L...,N...)
    espec::Array{Complex{Float64},D} = k2(L...,N...)
end
```

where we see a set of default parameters, and then some useful transform fields built using the parameters. Note that the transforms have to be built after building `X,K`.

6 Imprinting a dark soliton

We found a ground state by imaginary time propagation. Now we can impose a phase and density imprint consistent with a dark soliton. We will use the solution for the homogeneous system, which will be a reasonable approximation if we impose it on a smooth background solution.

6.1 Dark soliton in homogneous system

6.2 Dark soliton in a harmonic trap