

# Bright solitons (código)

Alejandro y Maria  
(Dated: 25 de Julio, 2016)

## I. PROGRAMA Y MÓDULOS

Evolución temporal de un solitón brillante partiendo de la solución analítica de la ecuación de Gross-Pitaevskii en 1D y sin potencial externo, utilizando el método de *split operator*. El programa consta de un main, cuatro módulos y dos scripts para las animaciones.

- `gpe_bright_solitons.py` (main)
- `gpe_bs_parameters.py`
- `gpe_bs_utilities.py`
- `gpe_bs_evolution.py`
- `gpe_bs_plots.py`
- `animation_gif.gnu`
- `animation.gnu`

Para ejecutar el programa, hace falta tener los siete scripts listados arriba en el mismo directorio, tener instalado Python 2.7, y correr el script "main":

```
$ python gpe_bright_solitons.py
```

o equivalentemente,

```
$ python2.7 gpe_bright_solitons.py
```

si ambas versiones de Python están instaladas.

Se pedirá introducir algunas variables por pantalla, y para interrumpir el programa en cualquier momento `ctrl+c`.

### A. `gpe_bright_solitons.py`

En el programa principal (main) se llaman funciones que definen la función de onda inicial (solitón brillante), el potencial y la evolución temporal (a tiempo real). Se puede elegir si hacer plots y/o escribir en ficheros la función de onda y magnitudes derivadas.

### B. `gpe_bs_parameters.py`

Se definen los parámetros iniciales y otras magnitudes, que son constantes en todo el programa. Los inputs para el usuario son:

- Scattering length:  $a_s = \{-0.5 \times 10^{-5}, -1 \times 10^{-5}, -2 \times 10^{-5}\}$
- Tiempo final: `time_final`
- Velocidad inicial del solitón:  $v = 0, 1, \dots, 10$
- Altura de la barrera:  $h_b = a \cdot \frac{v^2}{2}$ , con  $a = 0, 0.5, \dots, 2$
- Anchura de la barrera:  $w_b = \{0.5\xi, 1.0\xi, 2.0\xi\}$

### C. `gpe_bs_utilities.py`

Se define la función de onda inicial a partir de la solución analítica:

$$\psi(z) = \psi(0) \frac{1}{\cosh\left(\frac{z-x_0}{\xi\sqrt{2}}\right)}$$

$$\psi(0) = \frac{1}{2}\sqrt{|gN|} \quad \xi = \frac{1}{\sqrt{2|gN||\psi(0)|^2}} \quad gN = 2a_s N$$

Las unidades de energía y frecuencia son:

$$E_\xi = \frac{\hbar^2}{m\xi^2} \quad \omega_\xi = \frac{\hbar}{m\xi^2}$$

con  $\hbar = m = 1$ .

Se pueden utilizar distintos potenciales externos:

- Ningún potencial.
- Trampa armónica:

$$V(z) = \frac{1}{2}\omega_{\text{ho}}^2 z^2$$

- Barrera cuadrada:

$$V(z) = \frac{h_b}{1 + \exp\{m(z - xb_r)\}} - \frac{h_b}{1 + \exp\{m(z - xb_l)\}}$$

donde  $h_b$  es la altura,  $m$  la pendiente, y  $xb_r$  y  $xb_l$  son la posición de la pared derecha e izquierda de la barrera, respectivamente.

Independientemente del potencial, se pueden encerrar el solitón en una caja de parámetros fijos:

$$V(z) = h_w - \frac{h_w}{1 + \exp\{m(z - x_w)\}} + \frac{h_w}{1 + \exp\{m(z + x_w)\}}$$

donde  $h_w$  es la altura,  $m$  la pendiente, y  $x_w$  la posición de las paredes de la caja.

Hay también otras funciones que calculan la energía para un instante de tiempo, los operadores, la norma, y la integral bajo la curva.

### D. `gpe_bs_evolution.py`

Se calcula la evolución temporal. Para calcular la evolución a tiempo imaginario, `Dt` ha de entrar como complejo. Si en cambio es real, se calcula la evolución a tiempo real.

Se escriben ficheros con la función de onda y otras magnitudes para distintos instantes de tiempo, además de un fichero con las energías en función del tiempo, si `write_ev` es 0. Si no, sólo se escriben las energías.

Se escribe también en un fichero la norma de la función de onda, y un fichero preparado para hacer una animación de la densidad en gnuplot. Ambos son prescindibles, y sirven únicamente para comprobar que el programa funciona.

Los plots que aparecen dan la convergencia de la energía, la evolución de la densidad, y la integral a un lado y otro de la barrera.

### E. `gpe_bs_plots.py`

Se definen funciones para hacer diferentes plots. Se utilizan: `plot_convergence` (convergencia de las energías) y `plot_wave_function` (integral a ambos lados de la barrera).

## F. Outputs

Si se escriben ficheros de la evolución (`write_ev=0`), se crea una carpeta que lleva por nombre los inputs del usuario, modificado para que sean números enteros.

Dentro de esta carpeta, se genera un fichero con todas las energías en función del tiempo y las integrales a cada lado de la barrera, y distintos ficheros para cada tiempo con la función de onda (y otras magnitudes).

## II. TIEMPO DE COMPUTACIÓN

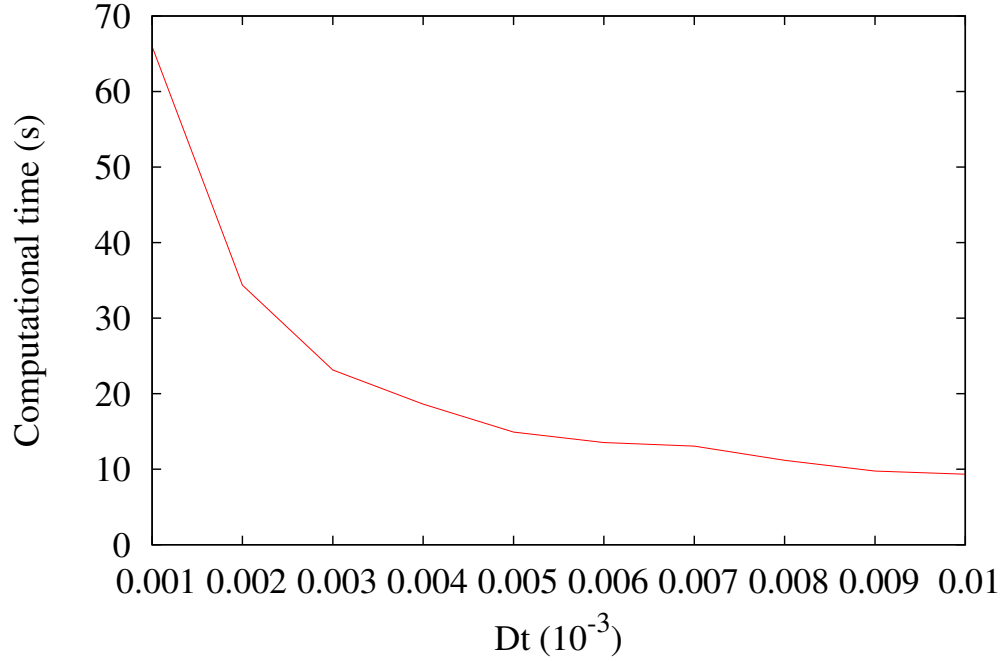


Figure 1: tiempo de computación en función del time step  $Dt$ .