

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE simulation [
<!ENTITY Nmodes      "64">
<!ENTITY Nsamples    "64">
<!ENTITY L   "3e-5">
]>
<simulation xmds-version="2">
  <name>GPE_1D_hermite_course</name>

  <author> Sebastian Wuester </author>
  <description>
    Gross-Pitaevskii-equation in 1D, SI units
  </description>

  <geometry>
    <propagation_dimension> t </propagation_dimension>
    <transverse_dimensions>
      <dimension name="x" lattice="&Nmodes;"
        length_scale="sqrt(hbar/(mass*omega))" transform="hermite-gauss" />
    </transverse_dimensions>
  </geometry>

  <features>
    <benchmark />
    <auto_vectorise />
    <validation kind="run-time" />
    <fftw />
    <globals>
      <![CDATA[
const double hbar = 1.05457266e-34;
const double omega = 10.0*(2.0*M_PI);
const double omega_perp = 200.0*(2.0*M_PI);

// Rb 87
const double mass = 1.4432e-25;
const double as = 5.5e-9;

//derived quantities
const double Natoms = 100.0;

const double U = 4.0*M_PI*hbar*hbar*as/mass;
const double sigma = sqrt(hbar/mass/omega);
const double sigma_perp = sqrt(hbar/mass/omega_perp);
const double U1d = U/(2.0*M_PI*sigma_perp*sigma_perp);
const double normfact = pow(M_PI*sigma*sigma,-0.25);

      ]]>
    </globals>
  </features>

  <vector name="wavefunction" initial_space="nx" type="complex">
    <components>psi</components>
    <initialisation>
      <![CDATA[
psi = 0.0;
if(nx==0)
psi = sqrt(Natoms);
      ]]>
    </initialisation>
  </vector>
```

```
<computed_vector name="moments" dimensions="" type="real">
  <components> norm expecxx expecx </components>
  <evaluation>
    <dependencies basis="x"> wavefunction </dependencies>
    <![CDATA[
      norm = mod2(psi);
      expecxx = x*x*mod2(psi);
      expecx = x*mod2(psi);
    ]]>
  </evaluation>
</computed_vector>

<sequence>
  <integrate algorithm="ARK89" interval="0.4" tolerance="1e-8">
    <samples>200 200 200 200</samples>
    <operators>
      <operator kind="ip" constant="yes">
        <operator_names>L</operator_names>
        <![CDATA[
          L = -i*(nx + 0.5)*omega;
        ]]>
      </operator>
      <integration_vectors>wavefunction</integration_vectors>
      <![CDATA[
        double dens=psi.Re()*psi.Re() + psi.Im()*psi.Im();

        dpside_dt = L[psi] - i*Uld*dens*psi/hbar;

      ]]>
    </operators>
  </integrate>
</sequence>

<output format="hdf5">
  <group>
    <sampling basis="x(&Nsamples;)" initial_sample="yes">
      <moments>density psire psiim </moments>
      <dependencies>wavefunction </dependencies>
      <![CDATA[
        density = mod2(psi);
        psire = psi.Re();
        psiim = psi.Im();
      ]]>
    </sampling>
  </group>
  <group>
    <sampling basis="kx(&Nsamples;)" initial_sample="yes">
      <moments>fspec</moments>
      <dependencies>wavefunction</dependencies>
      <![CDATA[
        fspec = mod2(psi);
      ]]>
    </sampling>
  </group>
  <group>
    <sampling basis="nx(&Nmodes;)" initial_sample="yes">
      <moments>energyspec</moments>
      <dependencies>wavefunction</dependencies>
      <![CDATA[
        energyspec = mod2(psi);
      ]]>
    </sampling>
  </group>
</output>
```

```
</sampling>
</group>
<group>
  <sampling basis="" initial_sample="yes">
    <moments> atomnumber meanpos deltapos </moments>
    <dependencies> moments </dependencies>
    <![CDATA[
      atomnumber = norm;
      meanpos = expectx/norm;
      deltapos = sqrt(expectxx/norm - (expectx/norm)*(expectx/norm));
    ]]>
  </sampling>
</group>
</output>
</simulation>
```