The Towler Institute

Quantum foundations workshop

21st-Century directions in de Broglie-Bohm theory and beyond

28th August - 4th September 2010

Vallico Sotto, Tuscany, Italy

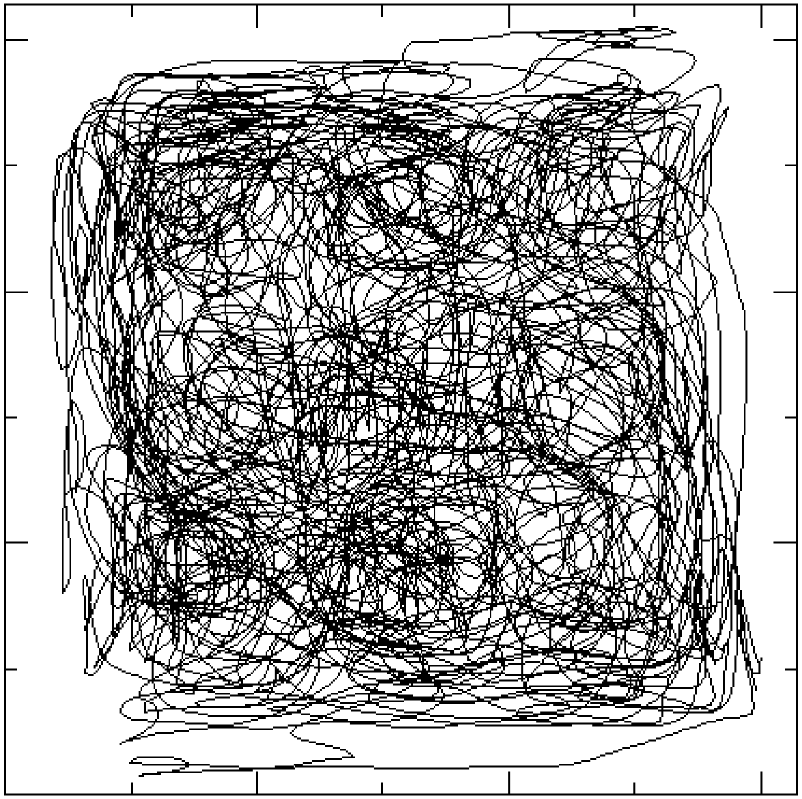
www.vallico.net/tti/tti.html

email : mdt26 at cam.ac.uk

Dynamical relaxation to quantum equilibrium..

..the magic of moving nodes

21st-century directions in de Broglie-Bohm theory and beyond, 30th August 2010



Mike Towler

TCM Group, Cavendish Laboratory, University of Cambridge

`www.tcm.phy.cam.ac.uk/∼mdt26` and `www.vallico.net/tti/tti.html`

`mdt26@cam.ac.uk`

First deBB research project

Study approach to 'quantum equilibrium' using numerical trajectories

(1) Try to extend work of Valentini/Westman [*Dynamical origin of quantum probabilities*, Proc. R. Soc. A 461, 253 (2005) - VW2005] and Colin/Struyve [*Quantum non-equilibrium and relaxation to equilibrium for a class of de Broglie-Bohm theories*, New J. Phys. **12** 043008 (2010) - CS2010] in some interesting way.

(2) Gain some experience in practical calculations (and write some sort of general computer program that is fast, easy-to-use, and readily extendable for more complicated problems).

(3) Try to figure out some interesting things to do in the future with this sort of technology.

Acknowledgements: Nick Russell, Antony Valentini

Quantum trajectory calculations

Schrödinger equation for both stationary and nonstationary states may be solved exactly by propagating quantum trajectories, at least *in principle*. Probability amplitude and phase of Ψ transported along trajectories and observables computable directly in terms of this information. Investigations that employ quantum trajectories may be broadly divided into two classes:

The analytic approach:

First solve TDSE using conventional techniques (fixed grids/basis set expansions). Individual 'particles' then evolved along quantum trajectories $\mathbf{x}(t)$ with velocities generated by Ψ-field via guidance equation $\dot{\mathbf{x}} = \frac{\hbar}{m}\,\mathrm{Im}\,\nabla\ln\Psi = \nabla S/m$. Patterns developed by trajectories emanating from ensemble of 'launch points' exactly define evolving system history. Used as means of understanding and exploring quantum behaviour i.e. point is not to solve TDSE but to provide *insight*.

The synthetic approach:

Rather than guiding quantum trajectories with a precomputed wave function, *the trajectories and wave function are computed concurrently, on the fly*. Wave packets are evolved by propagating ensembles of quantum trajectories, which become the *computational tool* for solving the 'quantum hydrodynamic' equations of motion.

Today we play around with a simple version of the first of these approaches.

Dynamical relaxation to quantum equilibrium. Why $\rho = |\Psi|^2$?

Pauli objection: Taking a particular particle distribution $\rho = |\Psi|^2$ as an initial condition is unjustified in a fundamentally deterministic theory, therefore 'theories' of this kind are incorrect [in *Louis de Broglie: physicien et penseur* Festschrift, 1953].

Pauli was right in the sense that this should be *derived* from the dynamics, for QM truly to emerge as the statistical mechanics of an underlying deterministic theory.

Easy to show if $\rho(\mathbf{x},t) = |\Psi(\mathbf{x},t)|^2$ at any t it will always remain so under Schrödinger time evolution ('*equivariance*'). Can also show $|\Psi(\mathbf{x},t)|^2$ is the *only* distribution with this property i.e. 'quantum equilibrium' is unique [Goldstein, Struyve 2007]. It is analogous to thermal equilibrium $P = \frac{\exp(-H/kT)}{Z}$.

'*With deterministic hidden-variable theories the Born distribution should not be regarded as an axiom. It should be seen as dynamically generated, in the same sense that one usually regards thermal equilibrium as arising from a process of relaxation based on some underlying dynamics.*'

A quite general argument (due to Antony Valentini, 1992) for the relaxation $\rho \rightarrow |\Psi|^2$ may be framed in terms of an analogy with the *classical coarse-graining H-theorem*. One may also look at *numerical simulations*.

What was done in Valentini/Westman 2005 paper

Trajectory calculations for particle in a 2D box with initial $\rho \neq |\Psi|^2$ and Ψ as a finite out-of-phase superposition of N energy eigenstates.

Normally say get fast relaxation to equilibrium on coarse-grained level for large numbers of particles. Clear from these simulations that large N in fact not needed for relaxation to occur. Even for one particle, relaxation occurs rapidly if its Ψ is a superposition of even a modest number of energy eigenfunctions.

Relaxation occurs because ρ and $|\Psi|^2$ evolve like two fluids 'stirred' by same velocity field. Most efficient mixing found to occur in neighbourhood of nodes or quasi-nodes, where Ψ is small. These points move around inside box, rather like 'electric mixers' or stirring devices moving through a fluid, generating an efficient relaxation everywhere.

Typicality and other views

Opposing camps exist in DeBB world differing in approach to $P = |\Psi|^2$.

- **Bell**: 'It is *assumed* that the particles are so delivered initially by the source'.
- **Holland**: Lists the $|\Psi|^2$ distribution as one of four basic postulates of pilot-wave theory.
- **Dürr, Goldstein, Zanghì** *et al.*: $|\Psi|^2$ regarded as natural measure of probability or 'typicality' for initial configurations of whole universe (taking Ψ as the universal wave function), yielding Born rule for all subsystems at all times. [See Dürr and Teufel book for good discussion].

AV: 'incorrect and deeply misleading'; postulates about initial conditions should have no *fundamental* status in a theory of dynamics. This seems fair enough to me.

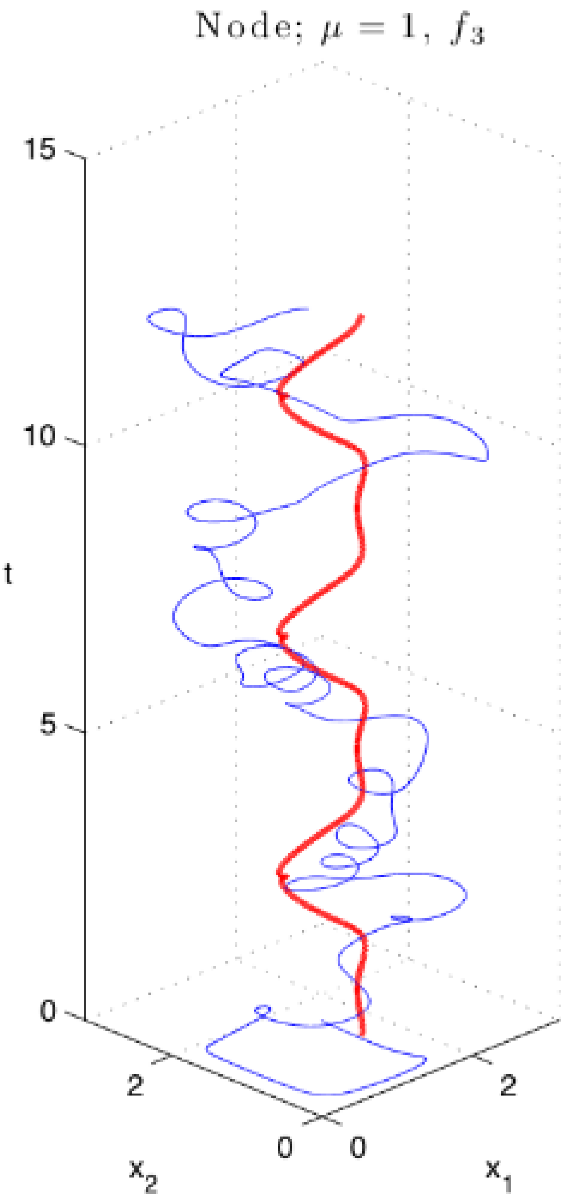
Additional simulations by Samuel Colin and Ward Struyve

'Quantum non-equilibrium and relaxation to equilibrium for a class of de Broglie-Bohm-type theories' (2010)

Repeat same 2D calculations while considering wider class of trajectory equations with alternative velocity fields which keep $|\Psi|^2$ invariant. Given $\mathbf{v} = \mathbf{j}/\rho = \nabla S/m$, can clearly add divergence-free term to current, i.e. $\mathbf{j} = \mathbf{j}_s + \mathbf{a}$ where $\nabla \cdot \mathbf{a} = 0$. Usually assumed $\mathbf{a}$ is zero (appropriate for spinless Schrödinger particles). For particles with spin, Lorentz covariance sufficient to determine relativistic particle law of motion uniquely. Assumption that non-relativistic guidance equation is a limit of the relativistic (Dirac) one then uniquely fixes that law too:

$$\mathbf{v} = \frac{\nabla S}{m} + \frac{\nabla \log \rho \times \mathbf{s}}{m} \quad \text{where} \quad \mathbf{s} = \frac{\hbar}{2}\chi^* \sigma \chi$$

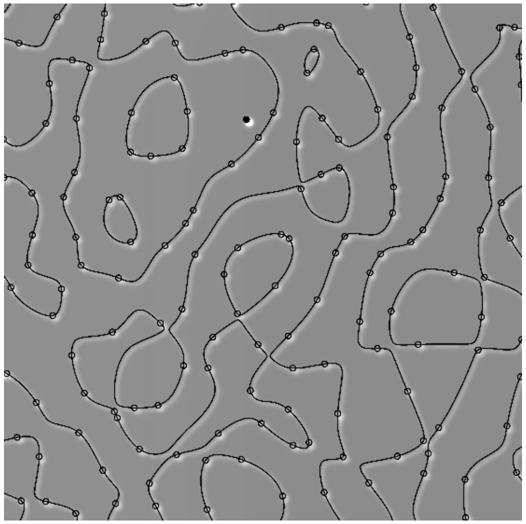
Spin thus a property of the wave field (the polarization-dependent part of its angular momentum) not of the particle. The spin affects the particle trajectory through a unique extra divergence-free contribution to the spin-independent Schrödinger momentum field.



- C+S found relaxation time depends substantially on form of guidance equation. In the regular Schrödinger theory the nodes are the only source of vorticity; the additional velocity field from the spin yields vorticity even away from nodes. Increased vorticity leads to greater chaotic behaviour and decreased relaxation times.
- Large percentages of trajectories could not be calculated - particularly those that started near nodes. Significant loss in accuracy in computing e.g. H function and relaxation times.

What I'm used to: nodes in quantum Monte Carlo

Nodes act as electric mixers! Not in QMC..



In fixed-node diffusion Monte Carlo calculations (the current state of the art) we are very preoccupied with nodal surfaces of wavefunctions. DMC accuracy depends only on shape of nodal surface of a 'guessed' trial many-electron wave function. We normally deal only with:

- Stationary states with real wave functions: Time-independent $3N-1$ dimensional hypersurface where Ψ is zero. Fixed-node approximation most significant error in DMC.
- Stationary states with complex wave functions: Less common but used in e.g. twist-averaging, non-collinear spins, magnetic fields. Fixed-phase approximation in DMC.

It is normally stated that e.g. "*very little is known about wave function nodes, and a systematic study has never been attempted*" (Bressanini, Ceperley, Reynolds) - meaning stationary state nodes as above. In this talk we make things easier for ourselves (not!) by dealing with .. *non-stationary* states with (necessarily) complex wave functions evolving in real time.

What do the nodes of such wave functions look like?

In general, we expect them to have a different topology to those for real functions. Imagine something like: for complex functions in 3D get nodes along lines where zero surfaces of real and imaginary functions cross. Similarly, in 2D lines become nodal points. Fewer barriers to movement of configurations? - important in 'time-dependent quantum Monte Carlo'? Save that for another time..

Student overhead: the time-dependent Schrödinger equation

Solve $i\hbar\frac{\partial}{\partial t}\Psi(x,t) = H\Psi(x,t)$ by separation of variables to give the following particular solutions (which have the counterintuitive property of predicting time-independent observables):

$$\Psi(x,t) = \phi_E(x)e^{-\frac{i}{\hbar}Et} \qquad |\Psi(x,t)|^2 = |\phi_E(x)|^2$$

Where has the time gone? It is restored to us by a general solution to the TDSE - an arbitrary superposition of the particular solutions:

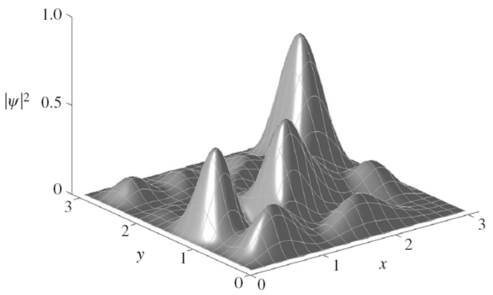
$$\begin{aligned} \Psi(x,t) &= \sum_{n=1}^{\infty} a_n\phi_n(x)e^{-\frac{i}{\hbar}E_n t} && \text{(discrete spectrum)} \\ &= \int_0^{\infty} a(E)\phi_E(x)e^{-\frac{i}{\hbar}Et}\,\mathrm{d}E && \text{(continuous spectrum)} \end{aligned}$$

Quite generally, a *wave packet* - a superposition of states having different energies - is required in order to have a time-dependence in the probability density and in other observable quantities, such as the average position or momentum of a particle. Simplest example: a linear combination of just two particular solutions $\Psi(x,t) = a\phi_E(x)e^{-\frac{i}{\hbar}Et} + b\phi_{E'}(x)e^{-\frac{i}{\hbar}E't}$. The probability density is given by:

$$|\Psi(x,t)|^2 = |a|^2|\phi_E(x)|^2 + |b|^2|\Psi_{E'}(x)|^2 + 2\mathrm{Re}\left\{a^*b\phi_E^*(x)\phi_{E'}(x)\mathrm{e}^{-i\frac{(E'-E)t}{\hbar}}\right\}$$

All the time-dependence is contained in the interference term.

The test case: particle in a 2D box



- System is a single particle in a 2D box with a (pure state) wave function $\Psi(x, y, t)$ satisfying Schrödinger equation ($\hbar = 1$)

$$i\frac{\partial \Psi}{\partial t} = -\frac{1}{2}\frac{\partial^2 \Psi}{\partial x^2} - \frac{1}{2}\frac{\partial^2 \Psi}{\partial y^2} + V\Psi.$$

- Box has sides of length π with infinite barriers. The energy eigenfunctions are

$$\phi_{mn}(x, y) = \frac{2}{\pi}\sin(mx)\sin(ny)$$

with energy eigenvalues $E_{mn} = \frac{1}{2}(m^2 + n^2)$, where $m, n = 1, 2, 3, \ldots$

- Initial *non-stationary* Ψ is superposition of first N modes ($m, n = 1, 2, 3, 4, \ldots$) with equal amplitudes but randomly chosen phases θ_{mn}. Note Ψ periodic in time with period 4π (since $4\pi E_{mn}$ is always an integer multiple of 2π).

$$\Psi(x, y, 0) = \sum_{m,n=1}^{\sqrt{N}} \frac{1}{\sqrt{N}}\phi_{mn}(x, y)\mathrm{e}^{i\theta_{mn}} \qquad \Psi(x, y, t) = \sum_{m,n=1}^{\sqrt{N}} \frac{1}{\sqrt{N}}\phi_{mn}(x, y)\mathrm{e}^{i(\theta_{mn} - E_{mn}t)}$$

Approach to equilibrium in classical statistical mechanics

For a **classical** *isolated system, both the probability density ρ and the volume element $\mathrm{d}\Omega$ (on phase space) are preserved along trajectories (Liouville's theorem).*

- Despite fact that $\mathrm{d}\rho/\mathrm{d}t = 0$ we find ρ evolves in a highly complex 'filamentary' manner over energy surface so *on a coarse-grained level* ρ becomes uniform as expected (whatever its initial shape). 'Coarse graining' involves dividing phase space into little cells of volume δV and working with *average* of ρ in each cell ($\bar{\rho}$).
- Can quantify difference between ρ and $\rho_{uniform}$ with *classical H-function*, i.e. $H_{class} = \int \rho \ln \rho \, \mathrm{d}\Omega$. This is minus relative entropy of ρ with respect to $\rho_{uniform}$ (standard measure of difference between two distributions). H bounded below by zero, and equals zero if and only if ρ uniform on energy surface (equilibrium).
- Classical $H_{class} = \int \rho \ln \rho \, \mathrm{d}\Omega$ is *constant in time*. If replace fine-grained ρ by coarse-grained $\bar{\rho}$ and assume $\bar{\rho}_0 = \rho_0$ at $t = 0$, then $\bar{H}_{class}(t) \leq \bar{H}_{class}(0)$ for all t - which is the **classical coarse-graining H-theorem**: i.e. $\bar{H}_{class}$ either decreases or remains constant, $\mathrm{d}\bar{H}_{class}/\mathrm{d}t \leq 0$. Decrease of $\bar{H}_{class}$ corresponds to formation of structure in ρ and consequent approach of $\bar{\rho}$ to uniformity.
- Relies on assumption $\bar{\rho}(0) = \rho(0)$ in phase space, i.e. no fine-grained microstructure in initial conditions (which could lead to 'unlikely' entropy-decreasing behaviour). Assumption necessary owing to time-reversibility of the theory.

Analogy with subquantum case if we let $\mathrm{d}\Omega \rightarrow |\Psi|^2 \mathrm{d}\mathbf{x}$ and ρ be the density of quantum particles.

Subquantum H-theorem

Valentini's argument for the relaxation $\rho \to |\Psi|^2$ is framed in terms of an analogy with the classical coarse-graining H-theorem.

- For 'sufficiently complex' system assume have initial distribution $\rho(\mathbf{x},0)$ of configurations $\mathbf{x}(0)$ each guided by the same Ψ, with $\rho(\mathbf{x},0) \neq |\Psi(\mathbf{x},0)|^2$.
- By definition $\rho(\mathbf{x},t)$ satisfies continuity equation $\partial\rho/\partial t + \nabla\cdot(\dot{\mathbf{x}}\rho) = 0$, and Schrödinger equation implies this is also satisfied by $|\Psi|^2$. Since $\dot{\mathbf{x}} = \nabla S/m$, clear that Ψ actually determines time evolution of ρ. So ratio $f = \rho/|\Psi|^2$ is preserved along trajectories: $\mathrm{d}f/\mathrm{d}t = \partial f/\partial t + \dot{\mathbf{x}}\cdot\nabla f = 0$.
- Initial deviations $\rho \neq |\Psi|^2$ thus forever carried along trajectories and never disappear, appearing to imply equilibrium unreachable (as with the ρ in classical stat mech). We now define the *subquantum H-function*: $H = \int |\Psi|^2 f \ln f \, \mathrm{d}\mathbf{x} = \int \rho \ln(\rho/|\Psi|^2)\, \mathrm{d}\mathbf{x}$. Continuity equation and $\mathrm{d}f/\mathrm{d}t = 0$ imply $\mathrm{d}H/\mathrm{d}t = 0$ i.e. exact fine-grained H constant as in classical case.
- Divide config space into cells of volume δV and define coarse grained-quantities e.g. $\bar{\rho} = (1/\delta V)\int_{\delta V} \rho \, \mathrm{d}\mathbf{x}$ etc.. For coarse-grained H have $\mathrm{d}\bar{H}/\mathrm{d}t \leq 0$; necessary and sufficient condition for $\bar{H}$ to have minimum value is $\rho = |\Psi|^2 \Rightarrow$ *equilibrium*. Decrease of $\bar{H}$ corresponds to a 'stirring' of the two 'fluids' ρ and $|\Psi|^2$ by the same velocity field $\dot{\mathbf{x}}$ (since satisfy same continuity equation), making ρ and $|\Psi|^2$ less distinguishable on a coarse-grained level.

See literature for e.g. defining quantum equilibrium of *subsystems*, and defining 'relaxation times'.

The birth of LOUIS

A new code to do quantum trajectory calculations and study the approach to quantum equilibrium



Comparison with previous code kindly sent to me by Samuel Colin.

- Heavier.
- Several orders of magnitude faster. Allows us to do *much* bigger calculations.
- More accurate - fixed a few numerical errors so ‘bad trajectories’ greatly reduced.
- Change behaviour of calculation using flexible input file rather than changing the source code. Introduced lots of internal error checking.
- General geometry.
- 1D, 2D, 3D calculations, not just 2D.
- Parallelized properly with MPI. Load balancing.
- Automated compilation with a proper Makefile and automatic architecture selection.
- Various modes of operation
  - single trajectory (forwards or backwards in time).
  - density
  - density + H function + relaxation time.
- Added timing routines
- Uses significantly less memory (‘allocate’).
- Velocity formulae selectable in input (e.g. ordinary deBB or inclusion of ‘spin term’).
- Add alternative integrators (Bulirsch-Stoer, etc.). Good consistency check.
- Random phase generator with fixed presets.

Things to do

- General wave functions (no analytic formula). Would require numerical integration of the Schrödinger equation.
- Multiple particles.
- Expanding space.

```
# LOUIS input file #

# BASIC
calc_type         : density               #*! trajectory or density (Text)
dimensionality    : 2                     #*! Number of space dimensions (Int)
int_algorithm     : runge-kutta           #*! Integration algorithm (Text)
timing_info       : T                     #*! Timers on/off (Boolean)
vel_type          : deBB                  #*! Velocity formula to use (Text)
curlweight        : 1.d0                  #*! Weight of vel eq. curl term (Real)
cell_x            : 3.141592653589793238  #*! Size of box: x (Real)
cell_y            : 3.141592653589793238  #*! Size of box: y (Real)

# TRAJECTORY OPTIONS
time_direction    : forward               #*! Forward or back in time (Text)
ntrajectories     : 1                     #*! No. of trajectories (Int)
traj_time_start   : 0.d0                  #*! Initial time (Real)
traj_time_end     : 12.566370614359172952 #*! Final time (Real)
%block trajectory_start
 1.570796326794896619 1.570796326794896619 0.d0

# DENSITY OPTIONS
den_ntimes        : 5                     #*! No. of density plots (Int)
den_time_start    : 0.d0                  #*! Time for first density plot (Real)
den_time_end      : 12.566370614359172952 #*! Time for last density plot (Real)
dentype           : 0                     #*! Select initial density (Int)
nlattice          : 1024                  #*! No. of lattice points (Int)
nscgrain          : 64                    #*! No. of points in scg cell (Int)
nsmoothstep       : 8                     #*! Step in smoothing procedure (Int)
read_backtracked  : F                     #*! Read backtrack positions (Boolean)
save_backtracked  : T                     #*! Save backtrack positions (Boolean)
plot_raw          : F                     #*! Plot raw densities (Boolean)
plot_cg           : F                     #*! Plot CG densities (Boolean)
plot_smooth       : T                     #*! Plot smooth CG densities (Boolean)
hfunction         : T                     #*! Compute H-function (Boolean)
plot_h_integrand  : F                     #*! Plot H integrand (Boolean)
fastmode          : T                     #*! Hard trajectories method (Boolean)
%block coarse_graining_lengths
 5
 4 8 16 32 64
```

```
# WAVE FUNCTION
wfn_type          : sine_wave             #*! Type of wave function (Text)
nmodes            : 4                    #*! Number of modes/phases (Int)
phase_format      : input                 #*! How to choose phases (Text)
phase_preset      : 0                     #*! Which set of preset phases (Int)
negphase          : T                     #*! Reverse sign of phases (Boolean)
transposephase    : F                     #*! Transpose 2D phase matrix(Boolean)
phase_noise       : 0                     #*! Phase noise in nth dp (Int)
%block phases
5.1305864145702568
6.2013294735531488
4.0145396815362080
3.4015356645345780
%endblock phases

# ACCURACY CONTROL
maxstp            : 100000                #*! Max integration iterations (Int)
init_eps          : 0.0001d0              #*! EPS tolerance initial value (Real)
init_h            : 0.0001d0              #*! Stepsize h initial value (Real)
converge_maxdiff  : 0.01d0                #*! Trajectory convergence tol (Real)

# DEVELOPMENT KEYWORDS
testrun           : F                     #*! Perform test run (Boolean)
verbose           : T                     #*! Toggle verbose output (Boolean)

# PLOT
plot_output       : x11                   #*! Type of plot output (Text)
plot_title        : auto                  #*! Title for plot (Text)
plot_type         : density               #*! Density/map/trajectory/bar (Text)
axislabels        : T                     #*! Add labels to axes (Boolean)
spacetime         : F                     #*! Spacetime traj in 2D (Boolean)
framerate         : 10                    #*! Framerate in fps - agif only (Int)
plane             : z                     #*! Plane of cross-section - 3D (Text)

# PLOT APPEARANCE
colour            : T                     #*! Toggle colour in plots (Boolean)
linegradient      : T                     #*! Line colour ==> z value (Boolean)
autorotate        : F                     #*! Gnuplot default rotation (Boolean)
rot_x             : 45.0                  #*! Rotation about x axis - 1st (Real)
rot_z             : 300.0                 #*! Rotation about z axis - 2nd (Real)
```

The code - what it's supposed to do

Assume 2D for simplicity. Start with some initial non-equilibrium particle density $\rho(x_1, x_2, 0)$ and some initial *non-stationary* wave field $\Psi(x_1, x_2, 0)$ (i.e. its absolute square is not constant in time). Calculate (coarse-grained) particle density $\bar{\rho}(x_1, x_2, t)$ at various t and compare its shape with that of the time-dependent wave field $\Psi(x_1, x_2, t)$. Expect particles to become distributed as square of the wave function over time. Monitor this 'approach to equilibrium' by computing the H-function.

How to evaluate the time-evolved density

Since $f = \rho/|\Psi|^2$ is constant along a trajectory, one can calculate the time-evolved density as:

$$\rho(x_1, x_2, t) = |\Psi(x_1, x_2, t)|^2 f(x_1(0), x_2(0), 0)$$

For neatness, accuracy, and pretty plotting we want the density to be on a regular grid at time t which is just what it *won't* be if we allow the trajectories to evolve from a regular starting grid of points. We therefore use a trick: we have a regular lattice of points defined on the cell at time t. We evolve these *backwards in time* using the de Broglie dynamics to random positions $x_1(0), x_2(0)$ at $t = 0$, at which points we evaluate the f function in the above formula. Hence $\rho(x_1, x_2, t)$ on a nice lattice.

How to evaluate the trajectories

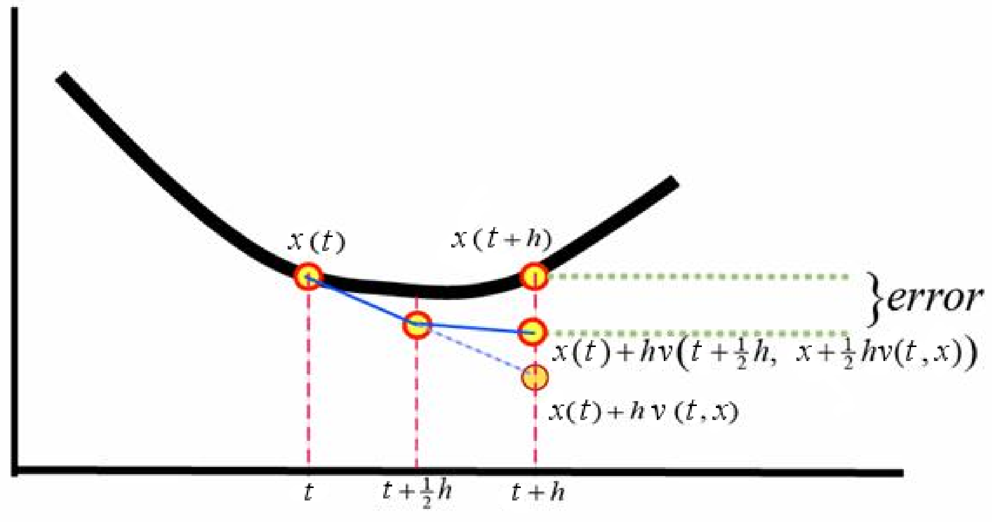
We need to integrate numerically the first-order ODE $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v} = \frac{\nabla S}{m} = \frac{1}{m}\mathrm{Im}\frac{\nabla\Psi}{\Psi}$, that is we shall effectively do the following (though hopefully with slightly more sophistication):

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(0) + \mathbf{v}\Delta t = \mathbf{x}(0) + \left(\mathrm{Im}\frac{\nabla\Psi}{\Psi}\right)\Delta t$$

Various possible algorithms e.g. Runge-Kutta, Bulirsch-Stoer, Verlet etc. Valentini *et al.* all used the *Runge-Kutta* method. We shall (at least initially) do the same.

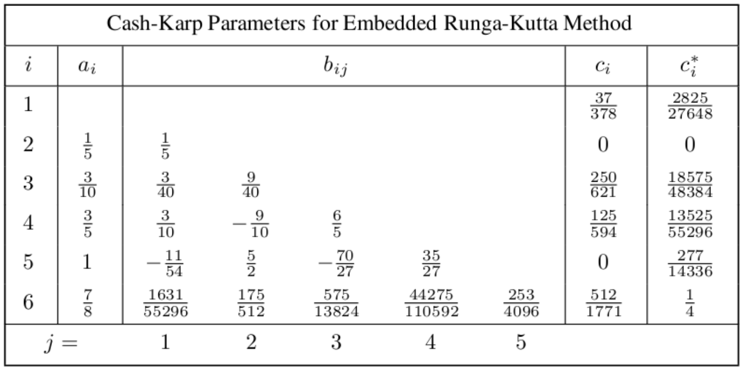
The Runge-Kutta method applied to calculating trajectories

- Literal implementation of $\mathbf{x}(t+\Delta t) = \mathbf{x}(0) + \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\Delta t$ gives the not-very-practical *Euler's method*. **Runge-Kutta** is cleverer, but it still comes down to adding small increments to your function given by derivatives ($\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} =$ velocity $\mathbf{v}$ in this case) multiplied by stepsizes (here, in time $\Delta t = h$).
- Runge-Kutta methods propagate a solution over an interval by combining the information from several Euler-style steps (each involving one evaluation of the derivative) and then using the information obtained to match a Taylor series expansion up to some higher order. The picture shows a 2nd-order algorithm (we actually use a fancy 5th-order one).



All you need remember is the general form of a 5th-order Runge-Kutta formula:

$$
\begin{aligned}
k_1 &= hv(t_n, x_n)\\
k_2 &= hv(t_n + a_2 h, x_n + b_{21}k_1)\\
&\cdots\\
k_6 &= hv(t_n + a_6 h, x_n + b_{61}k_1 + \cdots + b_{65}k_5)\\
x_{n+1} &= x_n + c_1k_1 + c_2k_2 + c_3k_3 + c_4k_4 + c_5k_5 + c_6k_6 + \mathrm{O}(h^6)
\end{aligned}
$$



Cash-Karp Parameters for Embedded Runga-Kutta Method

i	a_i	b_{ij}					c_i	c_i^*
1							$\frac{37}{378}$	$\frac{2825}{27648}$
2	$\frac{1}{5}$	$\frac{1}{5}$					0	0
3	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				$\frac{250}{621}$	$\frac{18575}{48384}$
4	$\frac{3}{5}$	$\frac{3}{10}$	$-\frac{9}{10}$	$\frac{6}{5}$			$\frac{125}{594}$	$\frac{13525}{55296}$
5	1	$-\frac{11}{54}$	$\frac{5}{2}$	$-\frac{70}{27}$	$\frac{35}{27}$		0	$\frac{277}{14336}$
6	$\frac{7}{8}$	$\frac{1631}{55296}$	$\frac{175}{512}$	$\frac{575}{13824}$	$\frac{44275}{110592}$	$\frac{253}{4096}$	$\frac{512}{1771}$	$\frac{1}{4}$
	$j =$	1	2	3	4	5		

So, essentially, every time you take a little step in t you have to compute the velocity six times. To get the velocity in a certain direction x you need both Ψ and $\nabla_x\Psi$, and the velocity is a hideous expression involving the sum of (say) 16 modes each of which involves two sines and a complex exponential. We're going to spend a lot of time calculating velocities.

For reasons too boring to discuss, the estimate of the error is $\Delta = \sum_{i=1}^{6}(c_i - c_i^*)k_i$. If we can estimate the error, we can try to keep it within desired bounds. If the error is too big we repeat the step with a smaller stepsize h: adaptive stepsize control.

As a further check, after computing the **entire trajectory**, we redo it with a smaller value for the desired maximum error. If the final positions of the two trajectories differ by more than some small amount, then we redo the whole thing with an even tighter tolerance. If we reduce the maximum desired error to some crazily small value and still two successive trajectories don't end up in the same place (or we exceed a large maximum number of RK steps), then we say the trajectory has **failed** and we discount it in our calculations of densities etc.. C+S had a big problem with this.

Mano a mano with Samuel Colin

The need for speed

They say the code is slow - let's make it go faster. Luckily the rate limiting step is quite obvious.. the velocity! For $1024 \times 1024 = 1048576$ points, a typical trajectory might take 3000 steps. Each call to a Runge-Kutta routine requires 6 velocities. Total number of calls to velocity routine to calculate density at one time step: around 22 billion. Adding a timer to CS2010 code confirms it - around 90% of the time is spent in the velocity routine. We must repeatedly evaluate $\mathbf{v} = \mathrm{Im}\frac{\nabla\Psi}{\Psi}$, where:

$$\Psi(x,y,t) = \frac{1}{\sqrt{mn}}\frac{2}{\pi}\sum_{m,n=1}^{4} \sin(mx)\sin(ny)\mathrm{e}^{i(\theta_{mn}-E_{mn}t)}$$

$$\nabla_x\Psi(x,y,t) = \frac{m}{\sqrt{mn}}\frac{2}{\pi}\sum_{m,n=1}^{4} \cos(mx)\sin(ny)\mathrm{e}^{i(\theta_{mn}-E_{mn}t)}$$

$$\nabla_y\Psi(x,y,t) = \frac{n}{\sqrt{mn}}\frac{2}{\pi}\sum_{m,n=1}^{4} \sin(mx)\cos(ny)\mathrm{e}^{i(\theta_{mn}-E_{mn}t)}$$

where $E_{mn} = \frac{1}{2}(m^2+n^2)$.

```
subroutine velocity(t,x,vnr,v)
use const,only:dp,dim,n,i,theta,pi,nv
integer, intent(in):: vnr
real(dp), intent(in):: t
real(dp), intent(in), dimension(dim):: x
real(dp), intent(out), dimension(dim):: v
real(dp),dimension(n)::cos1,cos2,sin1,sin2,vv
complex(dp) ::dfx,dfy,psi,phase

nv=nv+1
dfx = 0.0D0 ; dfy = 0.0D0 ; psi = 0.0D0
do k=1,n
 cos1(k)=cos(k*x(1)) ; sin1(k)=sin(k*x(1))
 cos2(k)=cos(k*x(2)) ; sin2(k)=sin(k*x(2))
end do
do k=1,n
 do l=1,n
  phase = 2.0D0*exp(-i*(theta(k,l)+ &
          & 0.5D0*t*k**2+0.5D0*t*l**2))/(n*pi)
  dfx = dfx+k*cos1(k)*sin2(l)*phase
  dfy = dfy+l*sin1(k)*cos2(l)*phase
  psi = psi+sin1(k)*sin2(l)*phase
 end do
end do
v(1)=aimag(dfx/psi) ; v(2)=aimag(dfy/psi)
select case(vnr)
   case(0)
   case(1)
     v(1)=v(1)+2.0D0*real(dfy/psi)
     v(2)=v(2)-2.0D0*real(dfx/psi)
   <snip>
end select
end subroutine velocity
```

Mano a mano with Samuel Colin

The need for speed

They say the code is slow - let's make it go faster. Luckily the rate limiting step is quite obvious.. the velocity! For $1024 \times 1024 = 1048576$ points, a typical trajectory might take 3000 steps. Each call to a Runge-Kutta routine requires 6 velocities. Total number of calls to velocity routine to calculate density at one time step: around 22 billion. Adding a timer to CS2010 code confirms it - around 90% of the time is spent in the velocity routine. We must repeatedly evaluate $\mathbf{v} = \mathrm{Im}\frac{\nabla\Psi}{\Psi}$, where:

$$\Psi(x,y,t) = \frac{1}{\sqrt{mn}}\frac{2}{\pi}\sum_{m,n=1}^{4} \sin(mx)\sin(ny)\mathrm{e}^{i(\theta_{mn}-E_{mn}t)}$$

$$\nabla_x\Psi(x,y,t) = \frac{m}{\sqrt{mn}}\frac{2}{\pi}\sum_{m,n=1}^{4} \cos(mx)\sin(ny)\mathrm{e}^{i(\theta_{mn}-E_{mn}t)}$$

$$\nabla_y\Psi(x,y,t) = \frac{n}{\sqrt{mn}}\frac{2}{\pi}\sum_{m,n=1}^{4} \sin(mx)\cos(ny)\mathrm{e}^{i(\theta_{mn}-E_{mn}t)}$$

where $E_{mn} = \frac{1}{2}(m^2+n^2)$.

How to improve this?

- Use a better compiler and turn on compiler optimization (!).
- Hand-optimize this routine (and a few others).
- Stop the velocity routine being called so much.
- Don't remove all the safety checks from the Runge-Kutta driver routine.

Time to calculate five densities on 1024×1024 grid

CS2010 code : 222791 sec (2.5 days) LOUIS : 13010 sec (3.6 hours)

Speedup: ×17.1. Samuel's cluster has 16 nodes, so we've saved him the cost of a parallel computer and he can now spend the money on girls and drink. (In some circumstances the speedup was much higher..).

```
subroutine velocity(t,x,vnr,v)
use const,only:dp,dim,n,i,theta,pi,nv
integer, intent(in):: vnr
real(dp), intent(in):: t
real(dp), intent(in), dimension(dim):: x
real(dp), intent(out), dimension(dim):: v
real(dp),dimension(n)::cos1,cos2,sin1,sin2,vv
complex(dp) ::dfx,dfy,psi,phase

nv=nv+1
dfx = 0.0D0 ; dfy = 0.0D0 ; psi = 0.0D0
do k=1,n
 cos1(k)=cos(k*x(1)) ; sin1(k)=sin(k*x(1))
 cos2(k)=cos(k*x(2)) ; sin2(k)=sin(k*x(2))
end do
do k=1,n
 do l=1,n
  phase = 2.0D0*exp(-i*(theta(k,l)+ &
          & 0.5D0*t*k**2+0.5D0*t*l**2))/(n*pi)
  dfx = dfx+k*cos1(k)*sin2(l)*phase
  dfy = dfy+l*sin1(k)*cos2(l)*phase
  psi = psi+sin1(k)*sin2(l)*phase
 end do
end do
v(1)=aimag(dfx/psi) ; v(2)=aimag(dfy/psi)
select case(vnr)
   case(0)
   case(1)
     v(1)=v(1)+2.0D0*real(dfy/psi)
     v(2)=v(2)-2.0D0*real(dfx/psi)
   <snip>
end select
end subroutine velocity
```

Accuracy problems

There is a connection between the following statements!

"If the calculation of the backtracked position involves more than 10^5 time steps, it is halted and the lattice point is ignored when calculating the coarse-grained and smoothed density."

"We have tried to perform the simulations . . . but the percentage of trajectories that could be backtracked was unacceptable."

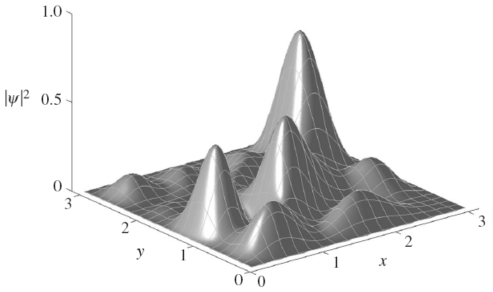
"Lattice points near nodes are generally harder to backtrack because the velocity field generally diverges there. This is in particular the case for the lattice points near the boundary of the box. Therefore in order to speed-up the calculations of the time-evolved non-equilibrium densities, we have ignored those lattice points that lie within two coarse-graining cells near the boundary. As such there is an error in calculating the coarse-grained $\bar{H}$-function.."

"Note that some of the values of $\bar{H}$ are negative. Theoretically this is impossible.."
Colin and Struyve (2010)

"The punishment for excessive greediness is interesting and worthy of Gilbert and Sullivan's Mikado: the routine can always achieve an apparent zero error by making the stepsize so small that quantities of order hy' add to quantities of order y as if they were zero. Then the routine chugs happily along taking infinitely many infinitesimal steps and never changing the dependent variables one iota. (You guard against this catastrophic loss of your computer budget by signalling on abnormally small stepsizes or on the dependent variable vector remaining unchanged from step to step. On a personal workstation you guard against it by not taking too long a lunch hour while your program is running."
Numerical Recipes in Fortran 77, 2nd edition (1992)

'Signalling on abnormally large stepsizes': computers can't add up 1.1826748282728462 and 1×10^{-18} !

Example test case: particle in a 2D box



- System is a single particle in a 2D box with a (pure state) wave function $\Psi(x, y, t)$ satisfying Schrödinger equation ($\hbar = 1$)

$$i\frac{\partial \Psi}{\partial t} = -\frac{1}{2}\frac{\partial^2 \Psi}{\partial x^2} - \frac{1}{2}\frac{\partial^2 \Psi}{\partial y^2} + V\Psi.$$

- Box has sides of length π with infinite barriers. The energy eigenfunctions are

$$\phi_{mn}(x, y) = \frac{2}{\pi}\sin(mx)\sin(ny)$$

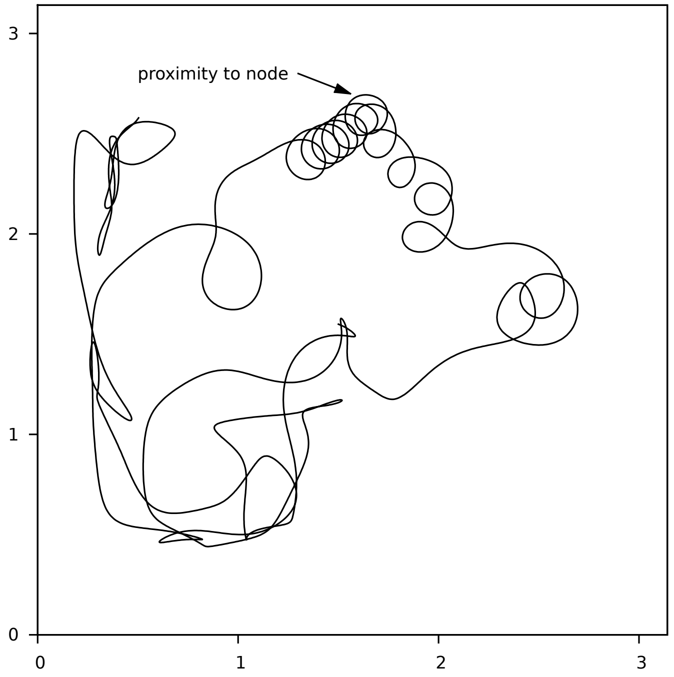
with energy eigenvalues $E_{mn} = \frac{1}{2}(m^2 + n^2)$, where $m, n = 1, 2, 3, \ldots$

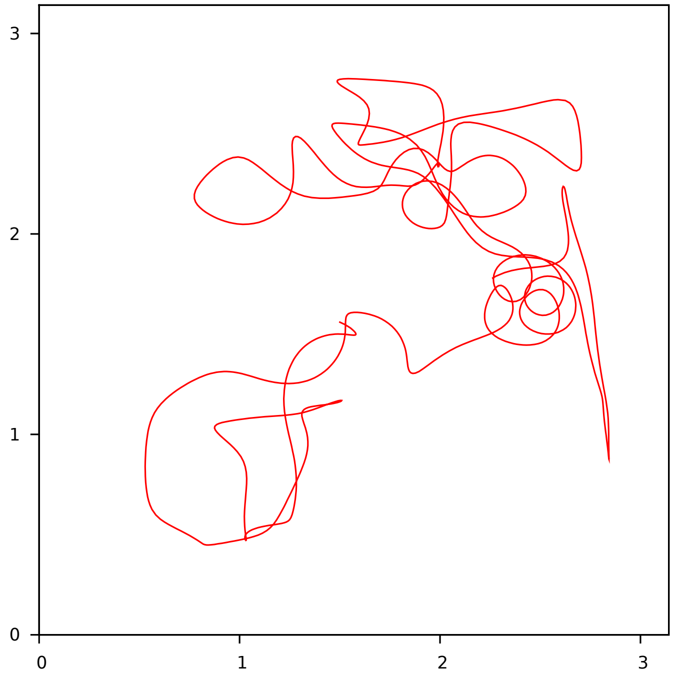
- Initial *non-stationary* Ψ is superposition of first N modes ($m, n = 1, 2, 3, 4, \ldots$) with equal amplitudes but randomly chosen phases θ_{mn}. Note Ψ periodic in time with period 4π (since $4\pi E_{mn}$ is always an integer multiple of 2π).

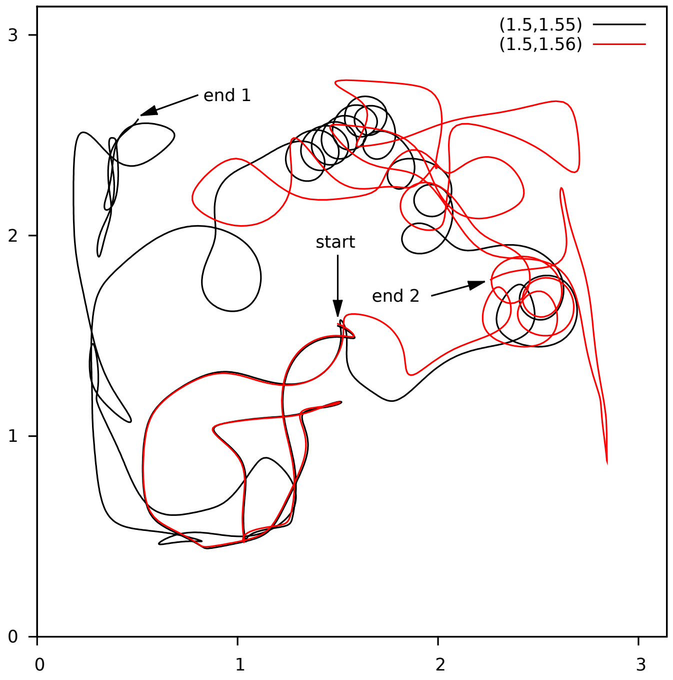
$$\Psi(x, y, 0) = \sum_{m,n=1}^{\sqrt{N}} \frac{1}{\sqrt{N}}\phi_{mn}(x, y)\mathrm{e}^{i\theta_{mn}} \qquad \Psi(x, y, t) = \sum_{m,n=1}^{\sqrt{N}} \frac{1}{\sqrt{N}}\phi_{mn}(x, y)\mathrm{e}^{i(\theta_{mn} - E_{mn}t)}$$

We have done these calculations with N=4, 9, 16, 25, 36, 49, 64 modes [previously only 16 (VW2005), or 4 (CS2010)]. Typical 36-mode calc 300 times slower than 4-mode calc - not possible before!

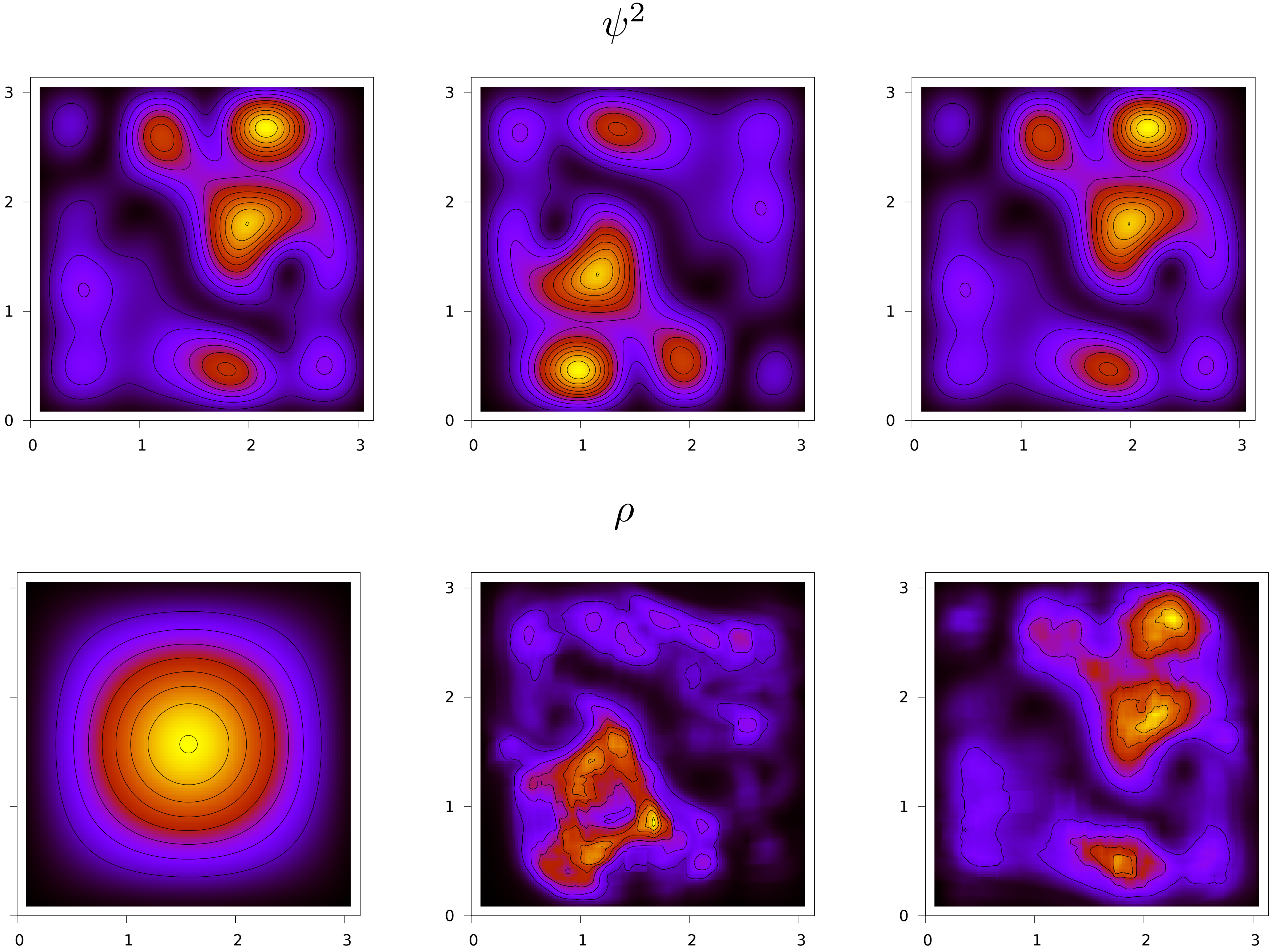
Examples of trajectories





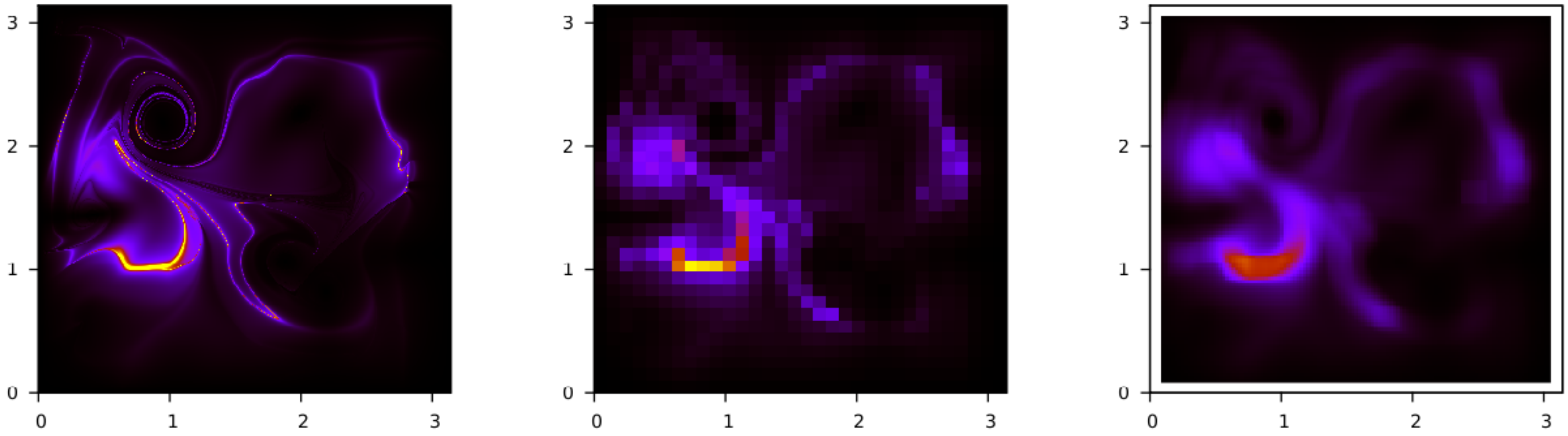


Dynamical relaxation to quantum equilibrium

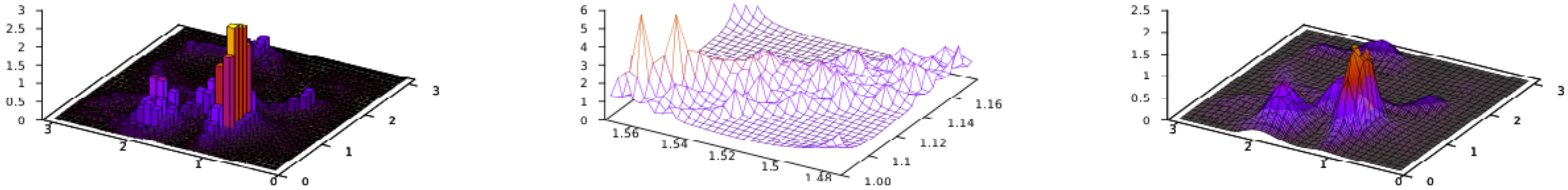


(a) t=0 (b) t=2π (c) t=4π

Effects of coarse-graining



(a) fine-grained (b) coarse-grained, $\varepsilon = 32$ (c) smoothed



(d) $\varepsilon = 32$ (e) single coarse-graining cell (f) smoothing applied

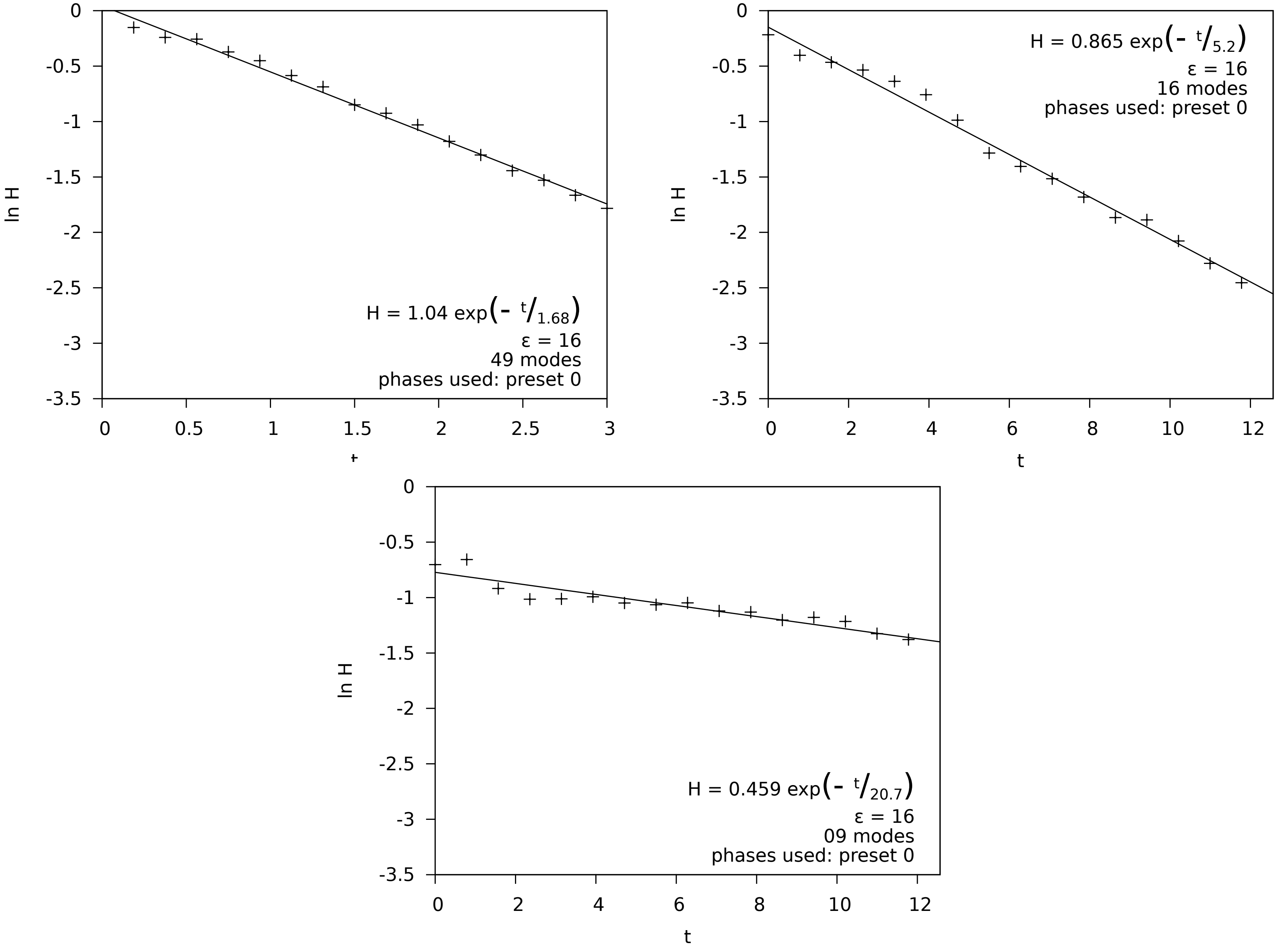
Effects of coarse-graining for a 16-mode system at $t = \frac{\pi}{2}$.

(d) ρ after coarse-graining of length 32 has been applied.

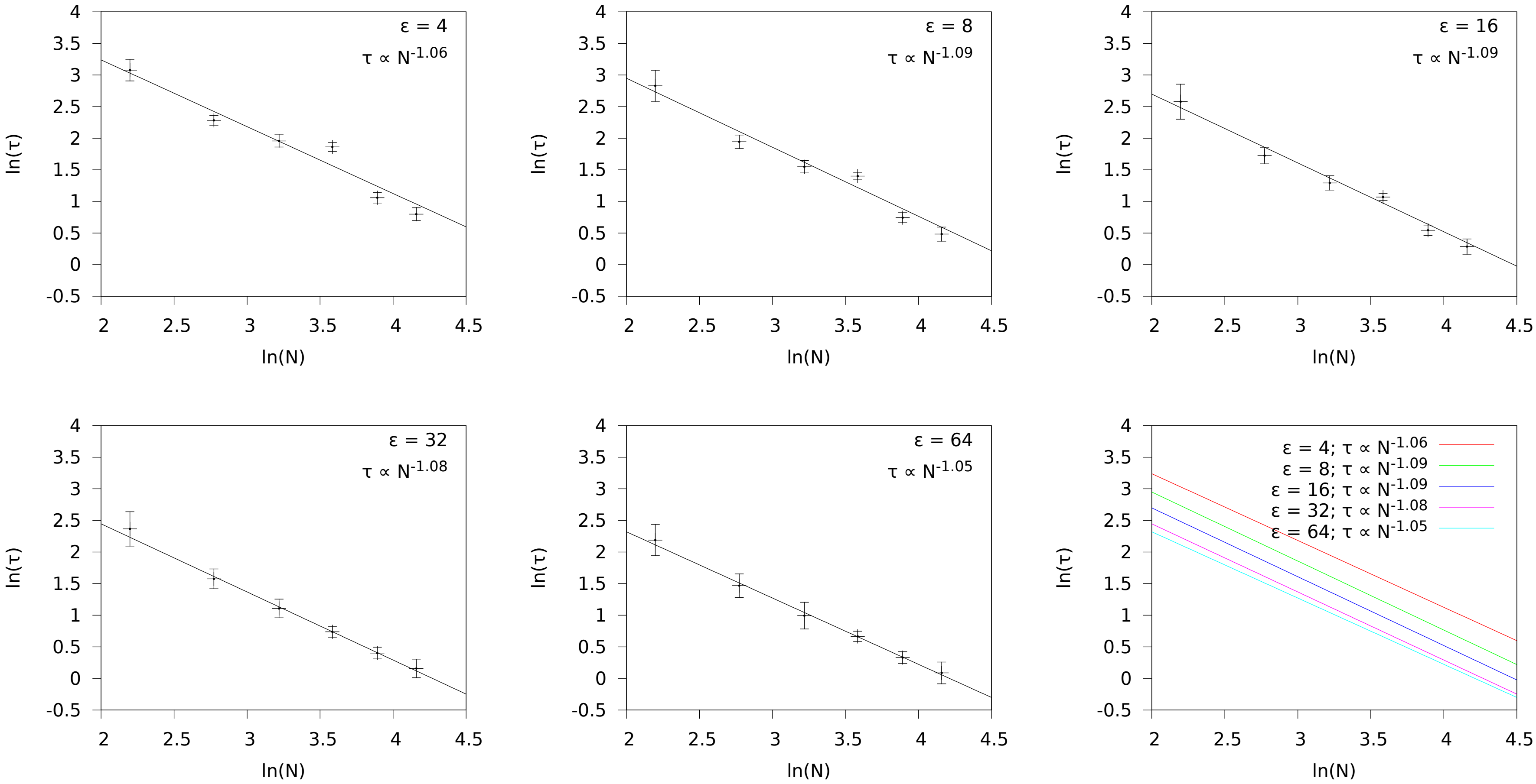
(e) Close-up on a single coarse-graining cell at the level of individual lattice points, showing the irregular nature of the underlying distribution.

(f) Effect of coarse-graining using overlapping cells, giving a smoothed distribution more suitable for visualization.

Exponential decay of the quantum H-function



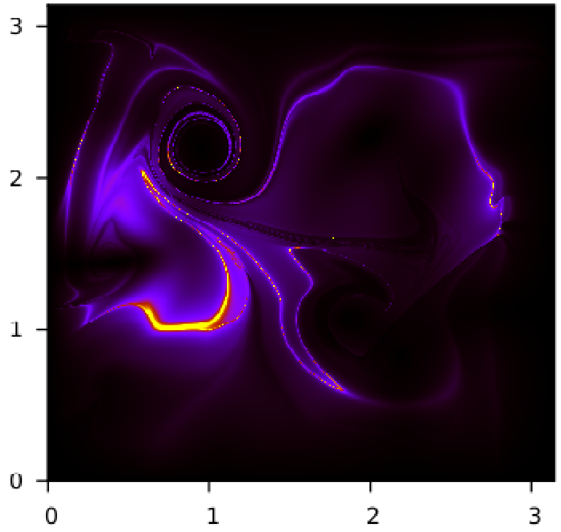
Power law fit for τ vs. N



- It can be seen that for all coarse-graining lengths there is a roughly constant power law with index around -1, in contrast to theoretical predictions of -3 (Valentini) (ΔE is the energy spread of the wave function and is proportional to N^2):

$$\tau \approx \frac{\hbar^2}{\epsilon m^{\frac{1}{2}} (\Delta E)^{\frac{3}{2}}}$$

Nodes



- Divergent velocity field in the vicinity of nodes drive the relaxation process - their initial distribution is very important in determining the relaxation rate. Varying the initial phases used in the wave function will change the position of the nodes.
- In large superpositions (large N) there are many nodes and their distribution may be approximated as uniform, so the average effect is similar for different sets of initial phases. With small superposition of e.g. 4 modes, this approximation will no longer be valid; there may e.g. be only 1 node and its initial position will have a much larger effect on the subsequent relaxation (it takes a lot longer and τ has big errorbar).
- Valentini's original prediction for the dependence of the relaxation time on the number of modes (N^{-3} power law) is faulty since it assumes that the velocity field varies little over the length of a coarse-graining cell. Especially with large numbers of modes in the superposition, this is not true.
- One can justify the observed N^{-1} scaling theoretically by defining a system to have relaxed when the mean displacement of a degree of freedom is greater than its quantum spread (see Antony's derivation in forthcoming Russell, Towler, Valentini paper).

Questions about nodal structure

- How many nodes are there in the 2D box? Is there a simple relationship between this and the number of modes in the superposition?
- In 2D, do the nodal points appear and disappear, or do they have a permanent existence? If not constant, what is the mean density of points in the plane?
- In 3D, do the nodal lines form loops or are they strings with ends? Can they be knotted?
- If the length in 3D varies, what is the mean length?
- How fast do the nodes move?
- Does pair-correlation for moving nodes depend on sense of vorticity of the surrounding current?

See work of Michael Berry: '*Phase singularities in isotropic random waves*'

'*Phase singularities, that is, dislocations of wavefronts - also called optical vortices - are lines in space, or points in the plane, where the phase of the complex scalar wave* $\Psi(\mathbf{r}, t) = \rho(\mathbf{r}, t) \exp[i\chi(\mathbf{r}, t)]$ *is undefined. For the generic smooth* Ψ *we are interested in, dislocations are also loci of vanishing* ρ*: in light, they are lines of darkness; in sound, threads of silence. Interest in optical dislocations has recently revived, largely as a result of experiments with laser fields. In low-temperature physics,* Ψ *could represent the complex order parameter associated with quantum flux lines in a superconductor or quantized vortices in a superfluid.*'

Calculates (1) mean length of dislocation line per unit volume, (2) mean density of dislocation points in the plane, (3) eccentricity of the ellipse describing the anisotropic squeezing of phase lines close to dislocation cores, (4) distribution of curvature of disclocation lines in space, (5) distribution of transverse speeds of moving dislocations, (6) position correlations of pairs of dislocations in the plane, with and without their 'topological charge'.

LOUIS: some ideas about what to do with it

Relaxation time τ

The relaxation time τ is the time over which $\bar{H}$ decreases (recall that $\bar{H} = 0$ then ρ and $|\Psi|^2$ are the same). Predicted theoretically to be inversely proportional to both the coarse-graining length ϵ and to $\Delta E^{\frac{3}{2}}$, where ΔE is the variance of the energy. Use LOUIS to study how τ varies with the number of modes/mean energy/energy variance numerically.

Do we get scaling by a simple power of energy variance? Or of mean energy? Is there a general, fairly robust scaling law, over some broad range of conditions? Would be useful to know, e.g. in cosmology.

Phenomenological equation for $\bar{H}$

Previous studies have found an approximately exponential decay of $\bar{H}$ with time. Can simulations suggest a phenomenological equation for $\bar{H}$ analogous to the Boltzmann equation? Presumably this behaviour could be derived by supplementing the underlying dynamics with some sort of phenomenological Markovian assumption, analagous to the classical hypothesis of molecular chaos at every instant.

Nodal studies

Is there a tendency for neighbouring trajectories to diverge especially rapidly when they pass near a node. Is the number of *nodes* a better measure of the relaxation rate? There might be a simple relationship in the limit of a large number of nodes (think of the nodes moving around in the box like a gas of moving particles, except each node is like an ‘electric mixer’ stirring up the ρ and $|\Psi|^2$ ‘fluid’ densities).

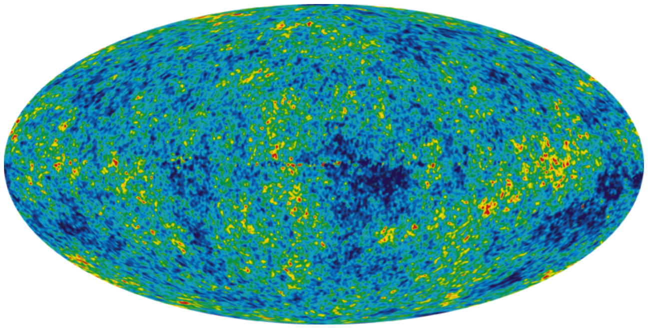
LOUIS: some more ideas about what to do with it

Effect of small perturbations

Can small perturbations drive relaxation over long timescales? E.g. take $\Psi(0)$ equal to the ground state plus excited modes with *tiny amplitude*. Then, $|\Psi|^2$ at all later times is close to the ground-state wavefunction-squared. Without the perturbations, an initial nonequilibrium ρ would remain static and always far from equilibrium. With the perturbations, can the trajectories eventually wander far enough to drive relaxation, or is the distance travelled forever too small? This would obviate the need for coarse-graining (necessary for isolated systems).

Look at expanding space

Valentini showed, for a scalar field on expanding space, relaxation to equilibrium is expected to be suppressed for a specific range of modes (long wavelength!). Possible consequence: a correction to predictions for the temperature fluctuations of the cosmic microwave background in the context of inflation theory.



Idea: generalize LOUIS so input is an initial nonequilibrium state for a given cosmology, and output is then a prediction of where nonequilibrium will be found later on. Expanding flat space is a good model of the early universe, and because of the expected asymptotic freedom all particles are effectively massless and ‘relativistic’ at high T, so using a free scalar field to model ‘matter’ isn’t a bad start. One might reasonably take the initial quantum state to be a mixed thermal ensemble of wave functions (Which basis to use for decomposing the density operator? There are some proposals..). The calculation can then be run separately for each pure sub-ensemble. LOUIS does this, going through all the different pure sub-ensembles, until it finds one that doesn’t relax. That would give a prediction for non-equilibrium today!

Finally MDT produces a de Broglie-Bohm paper!

Origin of the Born rule: timescales for dynamical relaxation to quantum equilibrium

M.D. Towler and N.J. Russell
TCM Group, Cavendish Laboratory, University of Cambridge,
J.J. Thomson Avenue, Cambridge CB3 OHE, U.K.

Antony Valentini
Theoretical Physics Group, Blackett Laboratory,
Imperial College, Prince Consort Road, London SW7 2BZ, U.K.

(Dated: October 7, 2010)

Abstract

We illustrate through explicit numerical calculations how the Born-rule probability densities of non-relativistic quantum mechanics emerge naturally from the particle dynamics of de Broglie-Bohm theory. The time evolution of a particle distribution initially *not* equal to the absolute square of the wave function is calculated for a particle in a two-dimensional infinite square potential well. Under the de Broglie-Bohm ontology, the box supposedly contains an objectively-existing wave field which influences the electron trajectory, and this is represented mathematically by a Schrödinger wave function composed of a finite out-of-phase superposition of N energy eigenstates (with N ranging from 4 to 64). The electron density distributions are found to evolve naturally into the Born-rule ones and stay there; in analogy with the classical case this represents a decay to 'quantum equilibrium'. The proximity to equilibrium is characterized by the coarse-grained quantum H-function which, as expected, is found to decrease roughly exponentially towards zero over the course of time. The timescale τ for this relaxation is calculated for various values of N and the length-scale of the coarse-graining ε. Its dependence on these quantities is found to be quite different to earlier theoretical predictions. A power law, $\tau \propto N^{-1}$ is found to be fairly robust for all coarse-graining lengths and, although a weak dependence of τ on ε is observed, it does not appear to follow any straightforward scaling. A theoretical analysis is presented to explain these results.