The analysis presented so far has been able to fully characterise the behaviour of the space-homogeneous kinetic model. However, it is unable to describe the dynamics of the full space-inhomogeneous model. It has been shown that the space-homogeneous model possesses only three invariant measures and converges to one of them dependent on the sign of the mean of the initial data. The full model also has these invariant measures, however it is not proven whether these are all of them or not. That is, the full model may have stationary distributions which are not homogeneous in space. By utilising numerical methods, we can relax some of the restrictions which the analysis requires. The goal is to show that for some choice of initial data and interaction function, the particle system will have a stationary distribution that isn't uniform in space. At its most basic, this could be two clusters of particles moving on the same direction diametrically opposite each other. The system will still converge in velocity to the values predicted by the homogeneous model, but spatial heterogeneity will remain. While this may be the case for the particle system, we conjecture that such behaviour will not occur in the kinetic model.

The aim is to be able to accurately simulate the dynamics of both the space-inhomogeneous PDE +++ref+++ and the corresponding interacting particle system. The latter is relatively simple to simulate using techniques for SDEs. However, we have shown analytically that the particle system does not have the same invariant measures as the continuum model, or indeed its McKean-Vlasov equation. The kinetic model is more difficult as it contains both advective and diffusive terms. We begin this section with a presentation of numerical methods for SDEs before reviewing methods for advection-type and diffusion-type equations, following the treatment of Hundsdorfer and Verwer [?] Hundsdorfer2007 as well as Morton and Meyers [?] Morton2005.

Mirroring the approach in the analysis, we will develop the techniques required first for the space-homogeneous system with no interaction, before adding further levels of complexity. Doing so allows the rigorous testing of any schemes developed as analytic solutions are available. Throughout this section it is recommended to follow along with the supplementary Jupyter notebook available at:

`https://github.com/Tom271/Whales` +++Change name+++

Every figure within this report was produced using the notebook and its accompanying source code.

0.1 Particle Systems

Simulating particle systems requires the numerical solution of coupled SDEs. The standard method for this is the Euler-Maruyama (EM) method, which can be seen as the stochastic analogue of Euler's method. Our motivating example here will be the Ornstein-Uhlenbeck process[a].

$$\mathrm{d}x_t = -x_t \,\mathrm{d}t + \sigma \,\mathrm{d}W_t \tag{1}$$

This is exactly the same evolution as that which drives the space-homogeneous particle system (??) eq:hom_particle, with no interaction between particles, and so provides the ideal starting point. It describes the motion of a damped particle moving randomly within a potential well as illustrated in Section ?? sec:particle_dynami.
Applying the EM method to the above equation gives:

$$x_{n+1} = x_n - x_n\Delta t + \sqrt{2\sigma\Delta t} Z_n,$$

[a]This is also the overdamped Langevin equation

Figures/OUparticletraj-eps-converted-to.pdf

Figure 1: Histogram of positions of 1000 particles after 100s with $\sigma = 1$, and positions of 5 particles over time. +++conv in moments?+++ articletraj

where Z_n is a standard normal random variable. This discretisation is very quick to compute, and has strong order $\frac{1}{2}$ [**?**] Higham01. This gives us a method of finding the stationary distribution: simulate many particles for a long time. Eventually, the density of the particles will be approximately the stationary distribution of the system. In this case it is in fact not even necessary to simulate many particles – if the system is allowed to evolve for long enough, one particle will suffice as there is no interaction between particles and the scheme is ergodic. The particle system will be an excellent metric by which to test the schemes developed for the continuum models. One drawback however will be the presence of only one invariant measure, as was shown analytically in Sec **??** sec:particle_dynamics and will be shown numerically in +++ref+++.

0.2 Diffusion Equations

As a prototypical example of a diffusion equation, consider the heat equation in one dimension given by

$$\begin{cases} \partial_t u(t,x) = \sigma \partial_{xx} u(t,x), & \sigma > 0 \\ u(0,x) = u_0(x), & t \in \mathbb{R}^+, x \in \mathbb{R}, \end{cases} \tag{2}$$ eq:heat

To solve this numerically, we must truncate the domain in both time and space. To do this, a region is chosen where we expect most of the mass to be throughout the time of interest. Here, we truncate to $x \in [-L, L], t \in [0,T]$ for some $L > 0$ and a finite time horizon T. Doing so requires a boundary condition to be enforced. A natural choice for this equation is a zero Dirichlet condition, $u(t,L) = u(t,-L) = 0$. As long as L is chosen large enough, the heat which spreads beyond this limit is negligible – this will however be a source of error. To mitigate this, one could compare the difference between analytic solutions (where available) on the full real line and the truncated domain, then choosing L accordingly. In general, boundary conditions are determined by the phenomenon being modelled. The problem geometry may naturally suggest conditions, or they may be enforced by numerical constraints as in this case. We must also discretise in space and time, to create a mesh covering $[0,T] \times [-L,L]$. Let $\{x_j\}_{j=0}^{J}$ partition the space equally such that $x_0 = -L, x_J = L$ and $x_j - x_{j-1} = \Delta x$ for all j. Similarly, let $\{t_n\}_{n=0}^{N}$ partition $[0,T]$ in such a way that $t_0 = 0, t_N = T$ and $t_n - t_{n-1} = \Delta t$. The parameters $\Delta x, \Delta t$ are the space step and time step respectively. We thus

Figures/FTCS.pdf

ig:FTCSmesh

Figure 2

have a discretised description of the continuum that we can implement.

Forward Time Centred Space

The aim is to solve the equation approximately on this grid. We shall denote approximate solutions by capital letters, that is $u(x_j, t_n) \approx U_j^n$. How can we approximate the solution? We must approximate each term in the equation separately. First the time derivative, or transient term can be approximated using the definition of the derivative. Recall

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{u(t+\Delta t) - u(t)}{\Delta t}.$$

We can then approximate the time derivative for fixed x by simply taking Δt small. Doing so leads to the forward difference approximation

$$\partial_t u(t_n, x_j) \approx \frac{U_j^{n+1} - U_j^n}{\Delta t}.$$

Now for the diffusion term, the most obvious approximation is to apply the forward difference scheme twice.

$$\begin{aligned} \partial_{xx} u(t_n, x_j) &\approx \partial_x \left(\frac{U_{j+1}^n - U_j^n}{\Delta x} \right) \\ &= \left(\frac{\partial_x U_{j+1}^n - \partial_x U_j^n}{\Delta x} \right) \\ &\approx \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2} \end{aligned}$$

This is sufficient to give a first approximation to (eq:heat 2).

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \sigma \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2} \tag{3}$$

eq:FTCS_hea

$$U_j^{n+1} = U_j^n + \frac{\sigma \Delta t}{(\Delta x)^2}[U_{j+1}^n - 2U_j^n + U_{j-1}^n] \tag{4}$$

This is the Forward Time Centred Space (FTCS) method, a fully explicit scheme which means that we can write the next time step solely in terms of known values. We now have a method for approximating the true solution. When developing any numerical scheme, three properties need to be considered: stability, consistency and convergence. These concepts will be developed further below. Heuristically, a scheme is unstable if certain mesh spacings cause rapid growth in solutions – a blow-up. A method is consistent if the error it makes in each step tends to zero as the size of the step tends to zero. Finally, a scheme is convergent if the approximate solution tends towards the true solution for any point in space and time.

The heat equation (eq:heat 2) can be solved by separation of variables, giving the solution as a Fourier series. If the initial data is a single Fourier mode, the solution is

$$u(t,x) = \mathrm{e}^{-\sigma(2\pi k)^2 t}\varphi_k(x), \qquad \varphi_k(x) = \mathrm{e}^{2\pi i k x} \text{ for } k \in \mathbb{Z}.$$

If we assume our approximate solution is a Fourier mode, a criterion for stability will emerge.

$$U_j^n = \lambda^n \mathrm{e}^{ik(j\Delta x)}, \qquad k \in \mathbb{Z} \tag{5}$$

eq:fourier_

$$\implies U_j^{n+1} = \lambda U_j^n, \qquad U_{j\pm 1}^n = \mathrm{e}^{\pm ik\Delta x} U_j^n \tag{6}$$

Clearly if $|\lambda| > 1$, the solution will be unstable. Substituting this into the finite difference scheme (eq:FTCS_heat 3) gives

$$\lambda = 1 - 4\mu \sin^2\left(\frac{k\Delta x}{2}\right), \qquad \text{with } \mu = \frac{\sigma \Delta t}{(\Delta x)^2}, \tag{7}$$

eq:FTCSlamb

and so,

$$U_j^n = \left(1 - 4\mu \sin^2\left(\frac{k\Delta x}{2}\right)\right)^n \mathrm{e}^{ikj\Delta x}.$$

Looking at (eq:fourier_mode 5), as $n \to \infty$, the numerical solution will grow unless $|\lambda| \leq 1$. If $|\lambda| \leq 1$, the solution will decay with higher modes being damped quicker. This is what we expect from our analytic knowledge of the heat equation. The restriction on the mesh spacing thus arises from (eq:FTCSlambda 7),

$$-1 \leq \lambda \leq 1 \implies \mu = \frac{\sigma \Delta t}{(\Delta x)^2} \leq \frac{1}{2}$$

This is not ideal: if more accuracy is required in the solution we can refine the space mesh, however in doing so the time step must also be shortened to maintain stability and the computational effort quickly increases.

The error made in each time step is the local truncation error, R, of the scheme. The local truncation error is the residual obtained by substituting the exact solution $u(t,x)$ into the discretisation.

Figures/stableFTCSheat-eps-converted-to.pdf

(a) $\Delta t = 0.005$

Figures/unstableFTCSheat-eps-converted-to.pdf

(b) $\Delta t = 0.0051$

Figure 3: Using the FTCS scheme to solve the heat equation (2) eq:heat on $[-5,5]\times[-1,1]$ with $\sigma = 1, \Delta x = 0.1$ with a Gaussian initial profile.

TCSunstable

Performing a Taylor expansion around $t = n\Delta t$ gives

$$\begin{aligned} u_j^{n+1} &= u_j^n + \Delta t \partial_t u_j^n + \frac{1}{2}(\Delta t)^2 \partial_{tt} u_j^n + \mathcal{O}(\Delta t^3) \\ \implies \frac{u_j^{n+1} - u_j^n}{\Delta t} &= \partial_t u_j^n + \frac{1}{2}\Delta t \partial_{tt} u_j^n + \mathcal{O}(\Delta t^2). \end{aligned}$$

And similarly around $x = j\Delta x$,

$$\begin{aligned} u_{j+1}^n &= u_j^n + \Delta x \partial_x u_j^n + \frac{1}{2}(\Delta x)^2 \partial_{xx} u_j^n + \frac{1}{3!}(\Delta x)^3 \partial_{xxx} u_j^n + \frac{1}{4!}(\Delta x)^4 \partial_{xxxx} u_j^n + \mathcal{O}(\Delta x^5) \\ \implies \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} &= \partial_{xx} u_j^n + (\Delta x)^2 u_{xxx} + \mathcal{O}(\Delta x^3). \end{aligned}$$

The local truncation error is then

$$\begin{aligned} R_j^n &= \underbrace{\partial_t u_j^n - \sigma \partial_{xx} u_j^n}_{=0} + \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2) \\ \implies R_j^n &= \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2). \end{aligned}$$

The first two terms are zero as u is solves the heat equation. The explicit scheme (3) eq:FTCS_heat is then first order in time and second order in space.

A numerical scheme is consistent if the local truncation error tends to zeros as $\Delta x, \Delta t \to 0$. The FTCS scheme is consistent of order one in time and two in space and stable so long as $\mu \leq \frac{1}{2}$. A scheme is convergent if for any fixed point $(x^*, t^*) \in [-L, L] \times [0, T]$,

$$x_j \to x^*, t_n \to t^* \implies U_j^n \to u(x^*, t^*).$$

Figures/BTCS.pdf

ig:BTCSmesh

Figure 4

The explicit scheme is convergent when it is stable. The stability condition is a great drawback of the FTCS method. In the next section we look to new methods to remove any restrictions on the mesh spacing, whilst maintaining (or improving) the truncation error.

Backward Time Centred Space

If instead of looking at the forward time difference we look at the backward time difference, the scheme becomes stable. This is surprising, very little appears to change but the scheme is fundamentally different. This method is implicit, meaning that its solution requires a more complex calculation at every time step. However, the lack of restriction on the size of the timestep means fewer steps (and therefore fewer calculations) need to be performed. The backward time centred space (BTCS) scheme for (eq:heat 2) is

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \sigma \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{(\Delta x)^2}. \tag{8}$$

eq:BTCS_hea

The only difference here is the right hand side is calculated at the $(n+1)^{\text{th}}$ time step whereas in the FTCS algorithm it is taken at n. This scheme cannot be written explicitly in terms of known values, that is, the n^{th} time step. Instead, we move all unknowns to the left hand side to give

where as before, $\mu = \frac{\Delta t}{(\Delta x)^2}$

```
Figures/CN.pdf
```

fig:CNmesh

Figure 5

Theta Method

To overcome the restriction in mesh spacing, one can use an implicit method, that is one where the solution depends on future time steps as well as the current. A method of this form is not as easy to solve, but as we will show here, has better accuracy and is unconditionally stable. The implicit Euler scheme appears the same as the explicit scheme above however here we calculate the spatial derivative at the future time step in place of the current one as follows:

$$\frac{U_j^{n+1}-U_j^n}{\Delta t}=\sigma\frac{U_{j+1}^{n+1}-2U_j^{n+1}+U_{j-1}^{n+1}}{(\Delta x)^2}$$

The Crank-Nicolson scheme involves taking the average of the implicit and explicit schemes outlined so far:

$$\frac{U_j^{n+1}-U_j^n}{\Delta t}=\frac{\sigma}{2}\left(\frac{U_{j+1}^{n+1}-2U_j^{n+1}+U_{j-1}^{n+1}}{(\Delta x)^2}+\frac{U_{j+1}^{n}-2U_j^{n}+U_{j-1}^{n}}{(\Delta x)^2}\right)$$

Rearranging this equation so that all unknown quantities (future time steps) are on the left hand side gives,

$$-\frac{\mu}{2}U_{j-1}^{n+1}+(1+\mu)U_j^{n+1}-\frac{\mu}{2}U_{j-1}^{n+1}=U_j^n+\frac{\mu}{2}\left(U_{j+1}^n-2U_j^n+U_{j-1}^n\right),$$

where $\mu=\frac{\sigma\Delta t}{(\Delta x)^2}$. This is a tridiagonal system and can be reduced to an upper triangular matrix and solved using the Thomas algorithm[b]. Doing so removes the costly requirement of inverting a matrix.

As for the stability of such a method, by applying the method seen for the fully-explicit scheme, one

[b]This is also known as the tridiagonal matrix algorithm

Figures/FTCSandCN-eps-converted-to.pdf

Figure 6: Solving the heat equation as in Figure fig:FTCSunstable 3 using `FTCS`, `BTCS` and `CN` respectively. The implicit solvers remain stable regardless of step size. g:ftcsandcn

obtains

$$\lambda = \frac{1 - 2\mu \sin^2\left(\frac{k\Delta x}{2}\right)}{1 + 2\mu \sin^2\left(\frac{k\Delta x}{2}\right)}.$$

Here $|\lambda| \ngtr 1$, so the method is unconditionally stable. We can also apply the same method as before to find the truncation error. Doing so gives that the Crank-Nicolson scheme is second-order in both time and space.

+++table summary of all methods (error order, stability) CN == Goldilocks.+++

0.3 Advection Equations

Our prototypical example in this case will be the one-way wave equation,

$$\begin{cases} \partial_t u(t,x) + a\partial_x u(t,x) = 0, & \\ u(0,x) = u_0(x), & t \in \mathbb{R}^+, x \in \mathbb{R}. \end{cases} \tag{9}$$

eq:wave

Figures/advgaussian-eps-converted-to.pdf

Figures/advindicator-eps-converted-to.pdf

g:advection

(a) $U_0 \sim \mathcal{N}(0,1)$ (b) $U_0(x) = \mathbb{1}_{[-1,0]}(x)$

Figure 7: Solving equation (eq:wave 9) with the upwind scheme. Even with very smooth initial data, dispersive effects can be severe.

Using the same discretisation as for the time derivative in the diffusion equation, yields the first order upwind scheme,

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \begin{cases} a\frac{U_{j+1}^n - U_j^n}{\Delta x} & \text{if } a > 0 \\ a\frac{U_j^n - U_{j-1}^n}{\Delta x} & \text{if } a < 0 \end{cases}$$

If the sign of a is not taken into account, the scheme is unstable. For further details see Hundsdorfer2007 [**?**]. +++Error, modified equation showing dispersion, CFL CONDITION +++

0.4 Finite Volume Schemes

In Section sec:dynamics **??**, equations were closed on the moments of the distribution. From this, we know that $\dot{M}_0 = 0$, that is mass is conserved. This suggests that numerical schemes designed to conserve mass would be ideal for solving this system. One such method is a finite volume scheme, a generalisation of finite difference methods. First, we write the system in conservative form. This is an equation of the form $\partial_u = \partial_x(a(x)u)$. For the space homogenous model (eq:space_hom_PDE **??**), this corresponds to

$$\partial_t f_t = \partial_v \left[(v - G(\langle w \rangle_{f_t}))\, f_t + \sigma \partial_v f_t \right]. \tag{10}$$

eq:flux_spa

Using the same discretisation as in Section sec:numericalmethods **??**, we further introduce auxiliary points $v_{j\pm\frac{1}{2}} = \frac{1}{2}(v_{j\pm1} + v_j)$. Then within any cell (or volume), $\Omega_j = [v_{j-\frac{1}{2}}, v_{j+\frac{1}{2}}]$, we can calculate the average.

$$\bar{f}_t(v_j) = \frac{1}{\Delta v}\int_{\Omega_j} f_t(v)\,\mathrm{d}v = f_t(v_j) + \frac{1}{24(\Delta v)^2}\partial_{vv} f_t(v_j)$$

Furthermore, the change in the average density within the cell is equal to difference of the mass lost through either side of the volume, that is,

$$\Delta v \frac{\mathrm{d}\bar{f}_t(v_j)}{\mathrm{d}t} = a(v_{j-\frac{1}{2}})f_t(v_{j-\frac{1}{2}}) - a(v_{j-\frac{1}{2}})f_t(v_{j-\frac{1}{2}}),$$

where $a(v) = \left[\left(v - G(\langle w\rangle_{f_t(v)})\right) f_t + \sigma\partial_v f_t(v)\right]$. This can then be approximated using a finite volume scheme:

$$\frac{F_j^{n+1} - F_j^n}{\Delta t} = \frac{1}{\Delta v}\left[a(v_{j-\frac{1}{2}})F^n_{j-\frac{1}{2}} - a(v_{j-\frac{1}{2}})F^n_{j-\frac{1}{2}}\right]$$

This is the first-order upwind scheme in conservative (flux) form. As in the previous upwind scheme, the choice of spatial points at which to evaluate depends on the sign of the advection term. That is,

$$a(v_{j+\frac{1}{2}})F^n_{j+\frac{1}{2}} = a^+(v_{j+\frac{1}{2}})F^n_j + a^-(v_{j+\frac{1}{2}})F^n_{j+1},$$

where $a^+ = \max(a, 0), a^- = \min(a, 0)$. This scheme is still order 1, however it provides the ideal setting for developing higher order schemes. +++show agreement with FD, particle, show symmetry in error+++