NOTE

- Numerical continuation is a useful tool for deterministic systems
  - It allows us to see how equilibria and limit cycles change when we change a parameter
- Stochastic dynamics can't be studied with standard continuation
  - Standard methods can't track randomness
- This work tries to change that
  - 10s summary of the method:
  - Compute a covariance matrix around the deterministic equilibria
  - Use that to define an ellipsoid within which the sample paths probably lie
  - Track those ellipsoids over parameter changes

NOTES Chapter 1: Intro

Deterministic systems

- Determinisim means no noise or randomness; if we know the state now, and the equations, we know it forever more
- Equilibrium = time-invariant solution
  - Which means that its some state that if the system starts there, it'll stay there
  - A hanging pendulum, where it lies balanced pointing downwards
  - The same but unbalanced, pointing upwards; as long as it's not perturbed, it'll balance there forever
  - Examples include a chemical equilibrium, where forward and backward reactions balance, to give a constant amount of reactant and product
- Equilibrium state depends on parameters
  - For example, in a chemical system, adding more heat might shift the balance from lots of reactant / little product to lots of product / little reactant
- Continuation reveals that dependence
  - It shows us where equilibria move to, as parameters change
  - Lets us compute curves showing where in state space an equilibrium will lie, for some given parameter value

NOTES From deterministic to stochastic

- Very little work to extend continuation to SDEs
  - Most research is on simulation and integration
  - Some methods for tracking invariant measures
  - Moment map method, where statistics of the system (eg. mean state vector, variance) are tracked
- How does small noise change deterministic results?
  - Say we have a deterministic system; it's easy to perform a continuation analysis
  - This reveals stable, unstable equilibria, limit cycles, etc.
  - How does the stability of these sets change when we add a small amount of randomness into the system?
  - Stable equilibria may become 'metastable'; solutions tend to remain close to the equilibrium, but noise-induced transitions may occur between the different stable states in the system
- We can extend numerical continuation to track local information about metastable equilibria
  - The resulting algo can be applied either during continuation, or as a post-processing tool after a determnistic continuation has been performed
  - The paper proposes that it can extend further to nonstationary solutions, stochastic PDEs, stochastic DDEs; later papers actually do this [I THINK?]

NOTES Section 2: an analytical example

- Consider a noise-corrupted pitchfork normal form
  - A pitchfork bifurcation is where a single stable equilibrium splits into two stable equilibria; a third, unstable equilibrium then appears between them
  - Details aren't particularly important, but basically it's one way for a system to transition from monostability to bistability
  - The system acts like a ball in a double potential well; if the ball is the system state, it'll roll down into one of two potential wells and stay in the bottom, when no noise is present
  - To noise-corrupt it, we add a little bit of randomness into the vector field; the system is then given by the deterministic component which does most of the work to evolve the system state, plus some random component that'll make the state jump around a little
  - The resulting bistability is similar to eg. climate dynamics: iceball-earth is a stable state, whereby earth temperature is kept low by ice caps reflecting radiation back to space; hot earth is also a stable state, whereby no ice caps mean less reflection meaning hot earth; randomness comes from volcanoes, forest fires, etc

Noise-induced pitchfork dynamics

- Interesting noise-induced dynamics occur in the bistable region
  - The randomness gives the ball a chance to jump out of one potential well and into the other, when the random nudges all add up in such a way as to push it up and over the unstable state
  - This gives rise to lots of interesting effects such as coherence resonance, where adding more noise into a neruonal system can make it more orderly and well-behaved; or stochastic resonance, where adding noise into a neuronal system can make it more sensitive to some input signal
- For any initial condition, we will almost surely visit visit both potential wells in finite time
  - Let's start the system at some arbitrary initial conditions
  - Let's fix the μ at some point that ensures bistability
  - Let's fix the noise amount at any non-zero amount (ie. at least some noise)
  - Even with huge distances between the potential wells, starting somewhere way away from either well, and tiny noise applied, with probability 1 the ball will visit the bottom of both potential wells within finite time
- We seek a stronger result; what insights can we gain into the timescales of the stochastic transitions?
  - Stochastic transition is just the jump from one potential well to the other
  - Intuitively, this will take a long time with small noise and deep potential wells, and be very fast with large noise and shallow potential wells

A graphic example

- The top of the figure is the same deterministic bifurcation diagram as we saw before
- At the bottom, we have example time series
- The red lines show the locations of the equilibria of the deterministic system; black lines show how the system behaves in time
- For small μ, the ball jumps very frequently between the potential wells; transition timescale is small
- For medium μ, the ball jumps occasionally between potential wells; transition timescale is medium
- For large μ, no jumps are seen over the 1000 units of simulation time
- We can see from the time series that the system seems to cluster around the deterministic equilibria
- There's a region of high trajectory density
- In the fast-transitions case the high-density region is still there, but the regions for the top and bottom metastables actually overlap
- Interesting idea: define some notion of high-density region, and use the distances between them as a proxy for the transition timescale
- High-density regions are shown in blue and green on the top plot
- They move further appart as the bifurcation parameter increases

Studying stochastic transitions

- What methods can we use to study the high-density regions?
  - We need a way of defining them mathematically, before we can hope to study their parameter dependence
- Fokker-Planck equations are inefficient
  - Builds a probability density function for the system state, for all time
  - It can be used to describe the probability of the system being at some point x, at some time t, for some initial conditions
  - Inefficient!
  - This essentially solves the stochastic differential equation at every point in phase space, which may be computationally inefficient when we're only interested in the local metastability of equilibria
- can we find a more efficient way of defining them?
  - With small noise intensities, the stochastic dynamics are very similar to their deterministic equivalent, over small timescales
  - We have efficient methods for studying deterministic systems, so can we repurpose those to the small-noise case?
  - If we can, we can come up with some nice elegant efficient algorithms for continuing the high-density regions

Continuation of metastable equilibria

Can we find a more efficient way of defining high-density regions?

- Linearise the system about each equilibrium point
  - Assume that the system lies close to one of the deterministic equilibria
  - Replace the deterministic component of the system with a linear approximation
  - This gives us an Ornstein-Uhlenbeck process, for which we can calculate the variance
- Calculate the variance of the resulting stochastic process
  - We can do this analytically, since we've linearised the original system
  - This gives us a simple formula for the variance of the system state around the equilibrium we linearised around
- Choose a ball around the deterministic equilibria, such that sample paths stay within it with high probability
  - The variance lets us determine how big this ball should be
  - That's much like confidence intervals, where we specify confidence in terms of standard deviations from the mean; data are fairly likely to be within 1 sigma, more likely to be within 2 sigma, etc.
  - Defining the ball size in terms of some variance multiplier (confidence level) therefore makes the 'dense region' definition consistent across processes, and interpretable though its conventional statistical analogs

Towards a stochastic continuation algorithm

We need three results to be able to use this definition in continuation

- Generalise variance ball construction to arbitrary-dimensional SDEs
  - Take our definition of a variance ball from before, and extend it away from the 1d case, towards something we can use for SDEs of any dimension
  - This forms section 3 of the work
- Efficiently compute the covariance matrix of the linearised SDE, at each point on the continuation curve
  - Need to find a way to exploit whatever information is already available from the continuation procedure
  - Forms section 4 of the work
- Define test-functions for overlapping stochastic neighbourhoods
  - Define some way of testing whether any given pair of ‘high-density / high-confidence’ neighbourhood balls overlap
  - Covered in section 5

Section 3: metastability and linearization

- We can linearise multidimensional processes easily
  - This just extends the 1-d case to processes with multiple state variables
  - Transform the variables so that the deterministic equilibrium is at the origin
  - Taylor-expand around the origin
  - Throw away the higher-order terms
  - We're then left with a multi-dimensional Ornstein-Uhlenbeck process
- The covariance matrix is then given by the solution to an ODE
  - I've skipped the full derivation because it's not very useful to anyone
  - Essentially we repeat the moments derivation from the one-dimensional case, but instead use a Jacobian matrix where previously we used scalar gadients
  - This would give us an integral equation, which we then differentiate to get an ODE for the covariance
  - The time-invariant solution of this ODE is the covariance matrix of the linearised SDE
- The time-invariant solution is a solution of a Lyapunov equation
  - As the equilibrium we linearised around is necessarily stable, the Lyapunov equation is guaranteed to be solvable
- Ellipsoids are defined with their principle axes scaled according to the inverse covariance matrix
  - As with the 1d case, sample paths will exist within these ellipsoids with high

Section 4: The Lyapunov equation

We have established the covariance matrix is given by the solution to a Lyapunov equation. How do we solve it for a single equilibrium? And for a branch of equilibria?

- Solution methods are well-studied within control theory
  - We can re-write the Lyapunov equation into a simple linear system of form Ax=b, and guarantee A is invertible
  - This means the system is definitely solvable, but perhaps there's a more computationally efficient way of approaching this than entirely restructuring the equation
- The continuation consideration adds several new aspects to the problem
  - We have access to the deterministic system Jacobian, as a result of the predictor-corrector steps of the continuation algorithm; this gives us one of the two necessary matrices
  - The other matrix can be found using at most one matrix multiplication; for additive noise, it can be precomputed without any preexisting knowledge
  - Since we're using the solution within parameter continuation, the covariance matrix at one parameter value is an excellent guess for the covariance matrix at another nearby parameter value. As such, we have excellent initial guesses for any iterative methods
- Covariance computation is actually fairly straightforward, with several methods available

Noise structure and degenerate ellipsoids

- If the covariance matrix is noninvertible, we can't define ellipsoids
- This can happen for certain system and noise structures
  - Say we have an uncoupled linearisation; each state variable evolves independently of the others, in the linearisation
  - If noise only acts on some state variables, there'll be some evolving stochastically, and some deterministically
  - In this case, the ellipsoid becomes ill-defined
- Define density neighbourhood over the stochastic variables only
  - Replace deterministic variables with their equilibrium value
  - Place the standard ellipsoid ball over the stochastic subset
  - Ellipsoid becomes well-defined again

Problem 3: Ellipsoids and test functions

- Distance between two ellipsoids indicates the timescale of their stochastic transitions; how do we compute it?
  - It's a useful problem in robotics, satellite control, computational geometry, etc.
  - Lots of different methods have been proposed as a result
- We choose a distance measure that doubles up as a test function
  - Test functions are just some equation that we plug data into, and it interprets the data for us
  - If the distance measure is >0, the ellipsoids are disjoint
  - Distance measure = 0 means ellipsoids just touch, at a point
  - Distance measure <0 means ellipsoids intersect
  - This allows us to test for dynamical switching at some timescale
- The distance is given by the solution to an optimization problem
  - I don't particularly understand the optimization step
  - Efficient algorithms are (apparently) available to solve it, and the paper outlines the key equations required to use them
  - As previously, we can use the previous step's solution as a good initial guess for the optimization procedure

Algorithm summary

Initialization step; for each equilibrium...

- Find a stable equilibrium of the deterministic component of the system
  - This is a standard problem. Could either solve though integration, or Newton's method
- Compute the linearisation of the deterministic system at that equilibrium
  - Basically just change the variables so that the equilibrium is at the origin, then take the Jacobian
- Set up the Lyapunov equation for covariances, and solve using Bartels-Stewart algorithm
  - This is the algo of choice as it doesn't require any initial guess

This gives us both an equilibrium to start the continuation from, and a covariance matrix which we can then re-use as an initial guess for subsequent solvings of the Lyapunov equation

The next step is to continue the equilibrium and covariance matrix under changes in a parameter

Algorithm summary

Iteratively...

- Take a predictor-corrector step, solving deterministic continuation equations at a new parameter value
  - This is the same as with standard continuation; predict the next solution value (equilibrium point + regularisation terms), then refine the prediction using root finding methods
- Iteratively solve the Lyapunov equation
  - Construct the Lyapunov equation for the current covariance matrix
  - Use the previous solution as an initial guess
  - Solve it iteratively, to get the current linearised covariance
- Construct a high-density ball, for some chosen confidence level
  - An ellipsoid whose shape is given by the covariance matrix, and size is given by the confidence level; will be smaller in directions with lower variance, larger in directions with higher variance
- Solve an optimization problem for the distances between each pair of balls
  - Use the distance between balls at the previous step as the initial guess for the current step

Outputs

- Deterministic equilibria
  - These are the standard equilibria that the system would settle to, if it weren't for random effects
  - Obtained by applying standard numerical continuation algorithms to the deterministic component
- Ellipsoids
  - These are the confidence balls around each equilibrium
  - They show where the sample path will very probably lie
  - They're basically the stochastic version of an equilibrium
- Mutual distances
  - The distance between pairs of ellipsoids
  - These give an estimate of the rate of stochastic transition between pairs of metastable equilibria
  - Big distances mean transitions are rare
  - Small distances mean transitions are more frequent
  - Zero or negative distance means transitions are very common

Example results

- Red, blue lines are stable equilibria of the deterministic system
- Red, blue circles are the high-confidence neighbourhoods, within which we expect the sample paths to lie
- Green line is an unstable saddle
- Fig b shows distance between ellipsoids; fig c shows mean stochastic transition rate; b predicts c very well
- Fig d squashes the main fig onto a plane
- Fig e,f are simulations of the system at the shown parameter value

Example results

- Shown here are the dynamics of a predator prey model
- The dynamics are complex, as the noise follows a correlated, multiplicative format
- The deterministic system also undergoes Hopf bifurcation
- The continuation algo still shows up useful results
- The ellipsoids touch the coordinate axis; if a trajectory reached here, a species would go extinct
- This stochastic justification is used to overcome a paradox in population models, whereby they cease to settle to an equilibrium when resources are sufficiently abundant
  - Justification is that stochastic effects mean a species would go extinct before that ever happened

Closing remark: a special case

- Ellipsoid separation is only a local heuristic for stochastic timescales
  - Local heuristic: the balls are defined from a linearisation around the equilibrium
  - The linear dynamics are only locally meaningful
  - Outside that locality, they might give us a heuristic description of the behaviours, but we can't read into it any more than that
- What if we could incorporate global information into the continuation?
  - How about instead of a local heuristic, we look for something more robust?
- Eyring-Kramer's law gives analytical switching rates in special cases
  - This relies on Jacobian computation
  - Jacobians are available from the predictor corrector step
  - We can therefore tag on the Eyring-Kramer's formula to get some additional information, that encompasses global dynamics
  - This is only available in the special case of a bistable gradient system