Qualitative behavior of random ordinary differential equations studied with polynomial chaos

Benjamin Sonday

Robert Berry

Habib Najm

Yannis Kevrekidis

Lots of other good people

Random ordinary differential equation

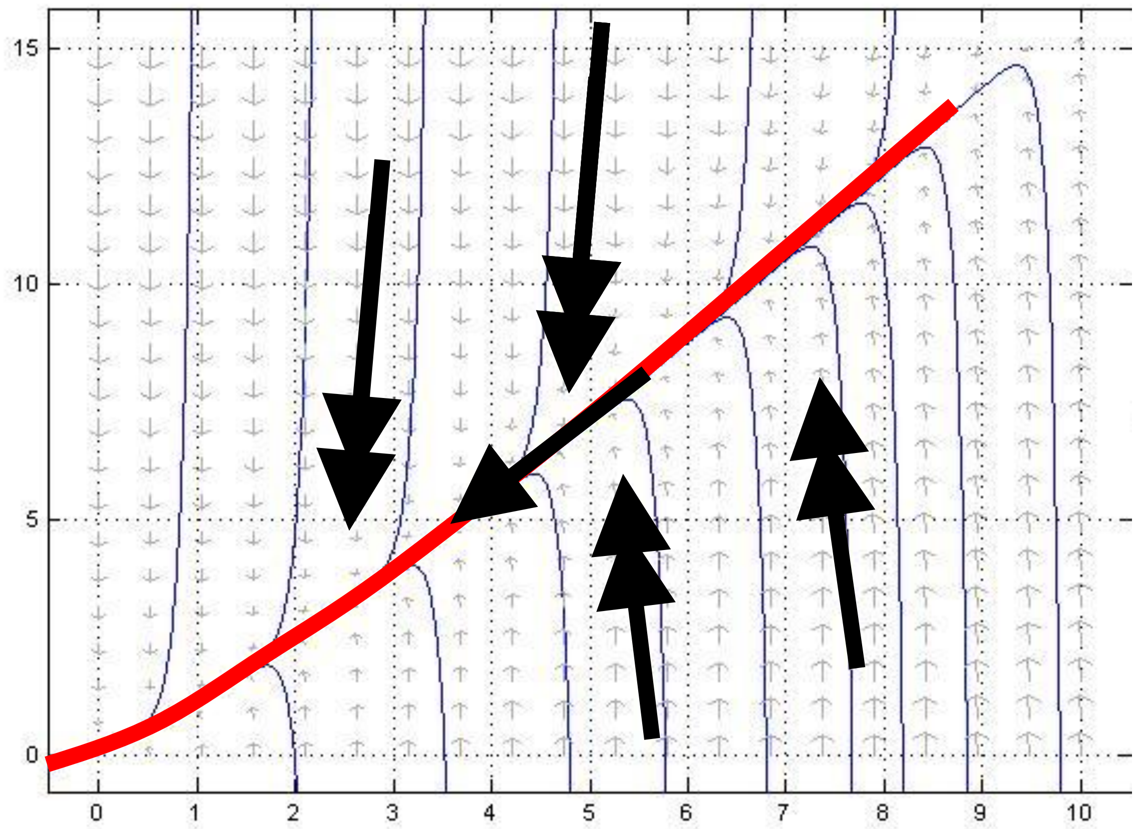
- Parameter
  - Reaction rate (chemical kinetics)
  - Velocity (Brownian dynamics)
- Initial condition
  - T=0 concentration of chemical species
  - Position: Heisenberg-type uncertainty
- Almost anything

Qualitative behavior

- Jacobian of the (random) ODE
- Eigenvalues
  - Dimension of slow manifold
  - Rate of decay to this slow manifold
- Eigenvectors
  - Position of slow manifold
  - Usually just a computational chore once the eigenvalues are known

Slow manifold

- “Fast directions” can be ignored
- Accelerates computation
- Provides intuition
- Reduces the dimension—usually, significantly



Description of problem

- Born out of a high-performance computational task
- Monte Carlo: random ODE gives a whole ensemble of
  - Jacobians
  - Eigenvalues
  - Eigenvectors
  - Can take a very long time…
  - How to quantify results?

Jacobian(ξ_1)
Jacobian(ξ_2)
⋮
Jacobian(ξ_M)

Polynomial chaos

- Like a Fourier basis for randomness—converges quickly!
- Easy to extract statistical information: mean, standard deviation, etc.
- Requires rewriting code (but not really)
- Turns random ODE into higher-dimensional deterministic ODE: N dimensional to NxP dimensional
- New Jacobian—expect the same, nice, low-dimensional behavior?

What we proved

- Eigenstructure of new PC Jacobian is basically the same as the old Jacobian
  - In essence, the eigenvalues of the new, NxD-dimensional Jacobian lie in the range of the old, Monte Carlo eigenvalues
  - Qualitatively, the same!
  - Proof applies to all Galerkin-type reformulations

Example

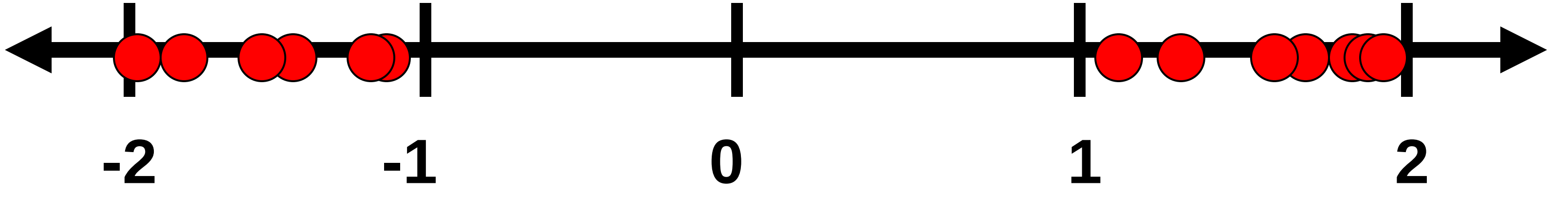
$$\dot{x} = ax$$

$$\frac{d}{dt}\, x(\xi, t) = a(\xi)\, x(\xi, t)$$

$a(\xi)$ is uniform on [-2,-1] U [1,2]

Example

Monte Carlo gives lots of
eigenvalues in [-2,-1] U [1,2]



Example

Polynomial chaos reformulation

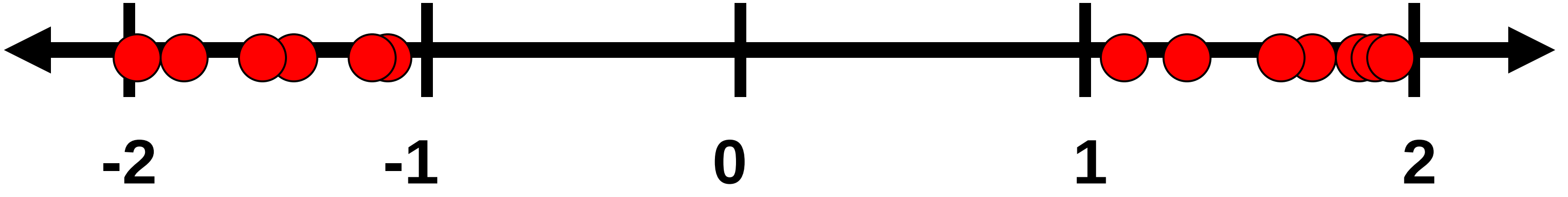
$$\dot{x} = ax$$

$$\sum_{k=0}^{P} \frac{\partial x_k}{\partial t} \Psi_k = \sum_{i=0}^{P} \sum_{j=0}^{P} x_i a_j \Psi_i \Psi_j$$

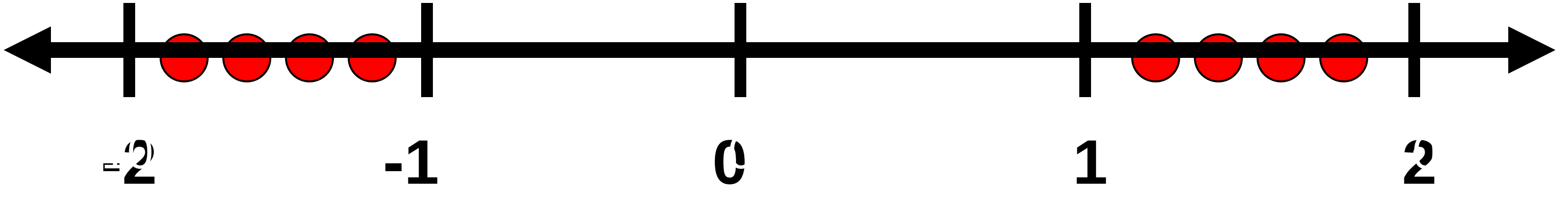
$$\frac{\partial x_k}{\partial t} = \sum_{i=0}^{P} \sum_{j=0}^{P} x_i a_j < \Psi_i \Psi_j \Psi_k > / < \Psi_k^2 >$$

Example

Monte Carlo gives lots of eigenvalues in $[-2,-1] \cup [1,2]$



Polynomial chaos gives the same



Takeaway points

- Polynomial chaos is
  - Fast
  - Informative
  - Easy
- Furthermore, it does NOT change the qualitative behavior of the system
- SIAM Journal of Scientific Computing article soon…

Acknowledgments

- CSGF/Krell
- Advisor
- Sandia Nat'l Lab
- Etc.

Today’s outline

- Motivation
- Description of problem
- Results
- Example