Complex Systems Simulation Models

Prof Ken Hawick

Massey University

December 2012

Discrete Lattice Models on meshes

- Ising
- Potts
- Sznajd
- Axelrod
- Conway Game of Life
- Wolfram CA
- Rock-Paper-Scissors-Lizard-Spock
- Invasion Percolation

(Regular) Field Models on meshes

- Cahn-Hilliard-Cook
- Time-Dependent Ginzburg-Landau
- Coupled Oscillators
- Lotka-Volterra & Variants
- Navier-Stokes & Variants
- Heisenberg

Particle Models in real-space (x,y,z)

- Gravitational Forces (Planets)
- Lennard-Jones Molecular Dynamics Forces
- Electromagnetic Forces (Plasma)
- Hard Disk/Sphere Collisions
- Variants - Hybrids; Mixtures; Repel/Attract
- Rule-based Systems - Boids

d-dimensions & geometry

- 1d - line
- 2d - plane (square, triangular, hexagonal)
- 3d - cubic or variants
- 4d and above - hyper-cubic

Neighbourhoods

- Nearest-Neighbour
- Next-Nearest-Neighbour
- Both = Moore neighbourhood
- Radial proximity - all within radial distance r

- Non-trivial counts in different dimensions

State-Space

- Can be finite
  - from discrete number of values on each site
  - eg 2^N for Ising on N-site lattice
- Nearly always too big to enumerate exactly

- Importance sampling
- Trajectory sampling

Initialisation

- Random uniform often used
- Nearly always not a likely state for the particular parameter combination being explored
- System relaxes towards a more likely state

Parameters

- Usually interested in some sort of macroscopic behaviour
  - eg order or disorder
  - Size of biggest cluster
  - Number of clusters
- Varies with microscopic parameters
- Or sometimes varies with Temperature

Time-Evolution

- Microscopic rule to update each site
- Explicit if-then-else rule
- Time-integration like finite difference
- Usually scheme is local - the inputs are:
  - The value of the cell itself
  - The value of its local neighbouring cells
- Sometimes all-to-all eg N-body particles

The Experiments

- Usually:
- Initialise randomly (hot start)
- Time-evolve (timestep or integrate)
- Once equilibrated (transients died out)
- Measure something that characterises the finalised or equilibrated system

- Do this lots of times to get average independent of any pathgologies from the particular starting values
- Do all over for different parameter values

Phase Transitions

- Often Find that there is some abrupt or unexpected change in the measured property (eg order parameter) even when control parameter is changed smoothly and slowly

- eg *Magnetization(Temperature)* in Ising
- *Largest-Cluster-Size(traits)* in Axelrod

Finite-Size effects

- Often find the location of the transition is affected by how big the system is
- Hopefully a weak but systemic effect

- Also usually find the location and nature of the phase transition varies with dimension and sometimes with lattice and geometry too.

How do we Simulate?

- Ideally want to be able to build software that:
  - Is Independent of dimension, geometry etc
  - Reuses hard-to-debug measurement code
  - Allows easy switch to a new & different model
  - Is Platform independent & portable
  - Is Reliable
  - Supports Graphical Visualisation
  - Is Very fast for batch runs and
  - Is Fast-enough for interactive runs (to debug & understand)

Choice of Languages

- C/C++/D/OpenGL
- Java/Swing
- Ruby/Python
- Sometimes must work with legacy eg Fortran
- Declarative eg Lisp/Scala,…

- Internal DSLs based on above
- Standalone External DSL

Libraries

- Sometimes language/System constrained by availability of a particular library
- eg scientific library code for
  - Matrix/linear algebra
  - BigNumber arbitrary precision
  - Rendering system
  - Arbitrary dimensionality - eg k-indexing
- Sometimes worth porting to desired platform

Complex Networks

- Some models defined on networks
- Meshes is just a regular graph
- Can add
  - Distortions
  - Extra/less connectivity
  - Long range (small world) connectivity
- Spatial proximities generate a graph
- So does geographic map data
- Or Internet/Power grids/bio-nets/…
- Other specialist graph generation algorithms

Static Properties

- Sometimes the static fixed properties of the graph itself are interesting
- Transitions or critical connectivities

- Sometimes focus on the model's dynamical properties
- Might behave different from mesh because of properties peculiar to the static graph

Dynamical Graphs

- Adaptive Meshes
- The network or graph itself changes with time
- Some algorithm or procedure for adding or removing nodes or arcs

- Dynamical properties need to be tracked in time

More Information

http://complexity.massey.ac.nz

K.A.Hawick@massey.ac.nz