

INTRODUCTION

COMPUTATIONAL MODELLING and SIMULATION

*Modelling and Scientific Computing
Research Group*

Speakers: H. Ruskin/L. Tuohey

1

- **HISTORY: 1953**
MANIAC simulates liquid !
- **Monte Carlo influence in Los Alamos**
- **Dynamic developments 1957,1959.....**

.....Computational Science.....!

2

CM &S. - Motivation

- **EXACTLY** soluble/Analytical Methods
- **IF NOT?** Approximate?
- **Simulation** → “Essentially” exact
 - “Essence” of problem
 - copes with intractability
 - can “test” theories and experiment
 - direct route micro → macro

3

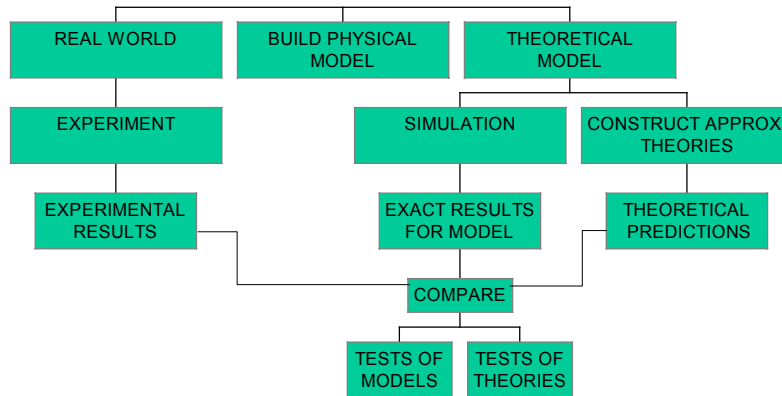
CM &S. - applications

- **Examples**
Atoms ... to galaxies, polymers, artificial life, brain and cognition, financial markets and risk, traffic flow and transportation, ecological competition, environmental hazards,

Nonlinear, Non-equilibrium...

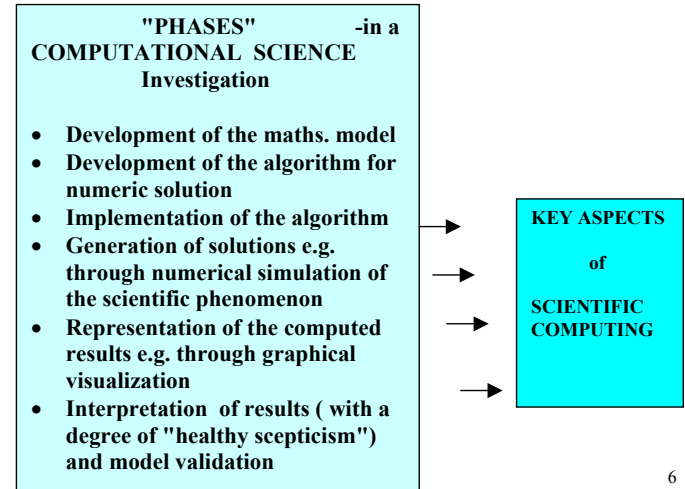
4

EXPERIMENT, THEORY & COMPUTER SIMULATION



5

CORE ACTIVITIES



6

COMPLEX SYSTEMS

- **Size** - *billions* of elements, events etc. *many variables*,
- **Changes** - *dynamics*
- **Lack of Sequence or pattern**
- **Instability** - (Non-equilibrium)
- **Non-constant cause-effect** (Non-linearity)
- **Global vs local changes**

etc. 7

COMPLEXITY 2

- WHAT SORT OF TOOLS?

- **Cellular Automata**- *"chess-board"*
- **Monte Carlo**- *random numbers*
- **Lattice Gas Models** - *Lattice-based, conservation laws* → *"Fluid" Models*
- **SOC** - *self-driven catastrophes*
- **Neural Networks** - *content addressable memory*
- **Genetic Algorithms**- *model evolution by natural selection: mutation selection*

8

MODELS - why do we need them?

- TO **PICTURE** HOW SYSTEM WORKS
- TO **REMOVE** NON-ESSENTIALS , called reducing the “*degrees of freedom*” (simple model first)
- TO **TEST IT**
- TO TRY **SOMETHING DIFFERENT**
.....*cheaply and quickly*

9

CATEGORIES of COMPUTATION

- **numerical analysis** (*simplification prior to computation*)
- **symbolic manipulation** (*mathematical forms e.g. differentiation, integration, matrix algebra etc.*)
- **simulation** (*essential elements - minimum of pre-analysis*)
- **data collection/analysis**
- **visualisation**

10

SIMULATION LEVELS

- **BRIDGING KNOWLEDGE GAP**
Idealised Model → Algorithm → Results
- **GOALS - SIMPLE LAWS**
 - PLAUSIBILITY
 - MEW METHODS/MODELS
- **DIRECT / INDIRECT**
- **PHENOMENOLOGY vs DETAIL**

11

NONLINEAR SYSTEMS

- **MOST Natural Phenomena - Nonlinear**
e.g. Weather patterns, turbulent flows of liquids, ecological systems = geometrical
- **CHAOS**
*e.g. unbounded growth or population explosion cannot continue indefinitely - **LIMIT** = sustainable environment*

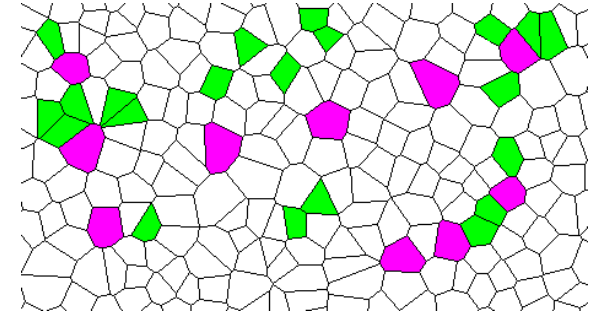
12

NON-EQUILIBRIUM SYSTEMS

- **Inherently UNSTABLE**
- **FEW THEORIES** - often simulation leads the way
- **EXAMPLE** - *foam coarsening*

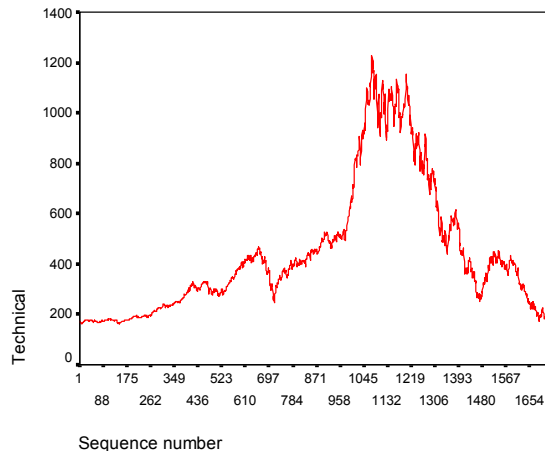
13

HOW FROTHS EVOLVE



14

EXAMPLE- Financial Markets, volatility/ turbulence



15

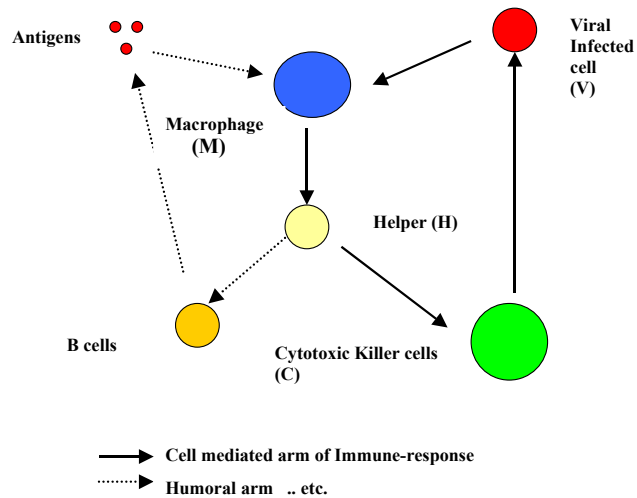
- Finance 2

- **KEY EVENTS**
 - 07/97 - 11/97 Roller-Coaster** → Asian Crisis
 - 14/09/98** -South American BAD news
 - 23/07/99** -plunge after highs/Greenspan address
 - 12/12/00** -U.S. Supreme Court judgement on Election Result
 - 28/03/01** -cut in Fed. Reserve rate not enough
 - 18/09/01** -post Sept. 11th

16

EXAMPLE

Immunological Response



17

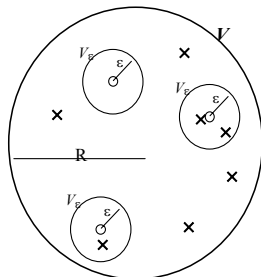
CURRENT- Immunology2

- **Physical Space** – Stochastic C.A
 - Microscopic
 - Macroscopic
- **Shape Space**
 - N-Dimensional Space that models affinity rules and repertoire size

18

SHAPE SPACE FORMULATION

introduced to predict repertoire size



Sphere of Influence for Immune System Components

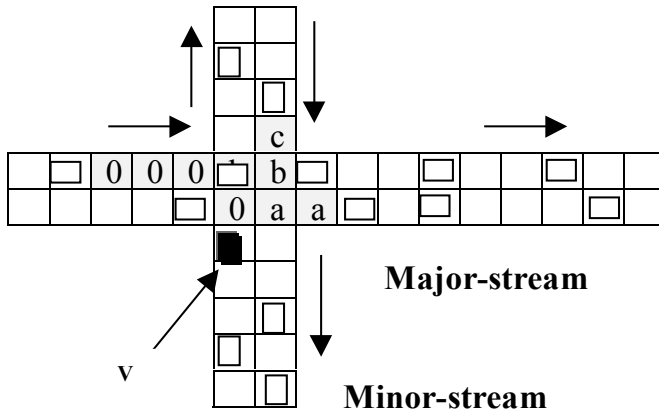
19

ALGORITHMS - MD, MC etc.

- **MD** - many particle - build dynamics from known interactions
- **MC** - most probable outcome (random numbers)
- **CA** - discrete dynamical systems. Finite states. Local updates.
- **FD/BV** - solns. D.E. 's - e.g. predictor/corrector algorithms

20

Example - TRAFFIC FLOW as a C.A. Model



21

Algorithms - contd

- **PROBLEM** - open-ended
 - **CORRECT/ENOUGH?**
 - Basics* - compare with known results
 - Orders of Magnitude
 - Errors/Limits
 - Extensions?
- *Coherent Story*

22

LANGUAGES ETC.

- **PROCEDURAL/ FUNCTIONAL, OBJECT-ORIENTED** - **Fortran, C** (change state or memory of machine by sequence of statements); **LISP, Mathematica, Maple** (function takes I/P to give O/P); **C++, JAVA** (Program = structured collection of objects)
- **PLATFORM**

23

NUMBERS, PRETTY PICTURES and INSIGHT

- **NUMBERS, PICTURES AGREEMENT?** - not enough

e.g. Simulation of river networks as a Random Walk. Path of Walker = Meandering of River

.....Why?

24