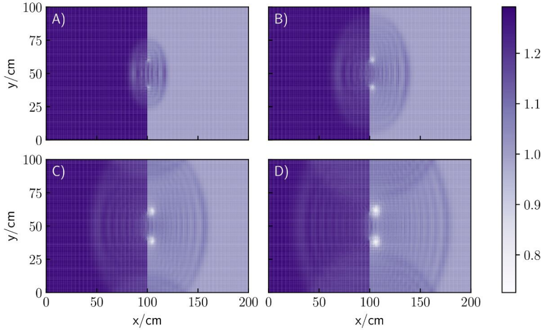
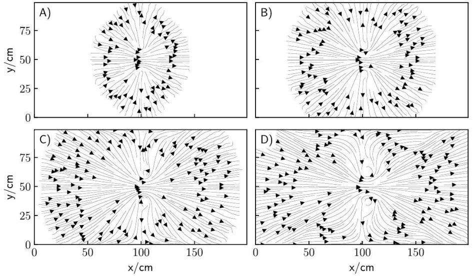
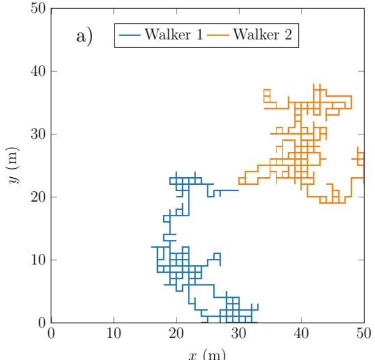
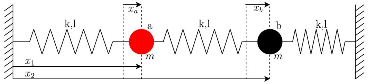
GreenPhysECS: Exploring ECS for Energy-Aware Parallel Research Software

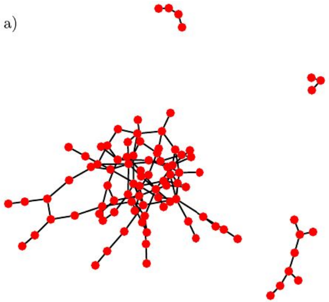
Oluwole Delano, Mattias Evans, Michael Sparks*
Physics and Astronomy, University of Manchester

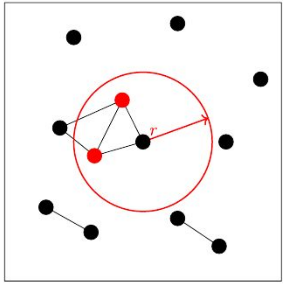










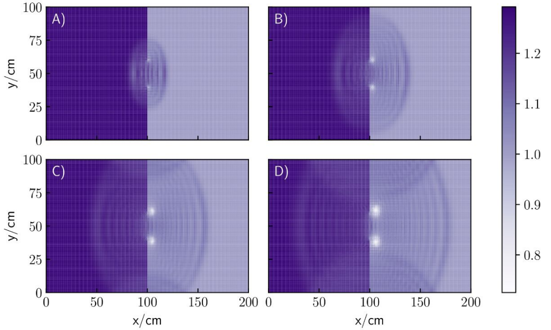


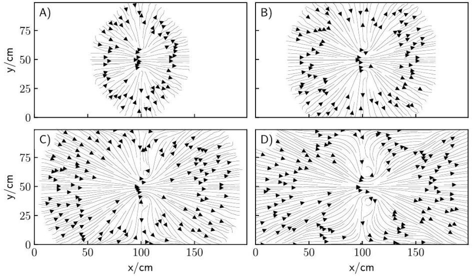
michael.sparks@manchester.ac.uk (future contact)

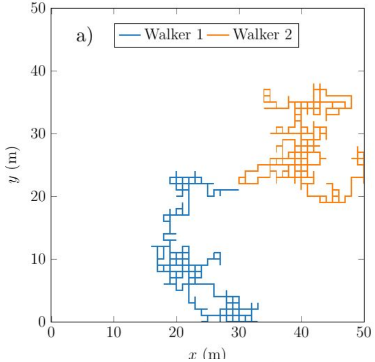
GreenPhysECS: Exploring ECS for Energy-Aware Parallel Research Software

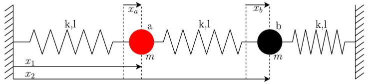
Oluwole Delano, Mattias Evans, Michael Sparks*

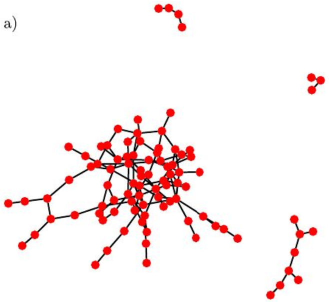
Physics and Astronomy, University of Manchester

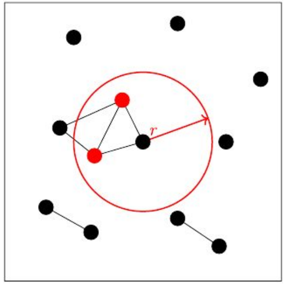












michael.sparks@manchester.ac.uk (future contact)

MANCHESTER 1824
The University of Manchester

Problem

- Physics Workloads are embarrassingly parallel
- Highly parallel hardware is now standard - **from laptops through clusters**

- **Software pedagogy, practice and tooling often remain linear - which does not map well onto such infrastructure**

- **Valuable research infrastructure can remain difficult to use effectively - especially for ECRs**

The problem is fundamentally around **enabling and widening good utilisation** of modern infrastructure

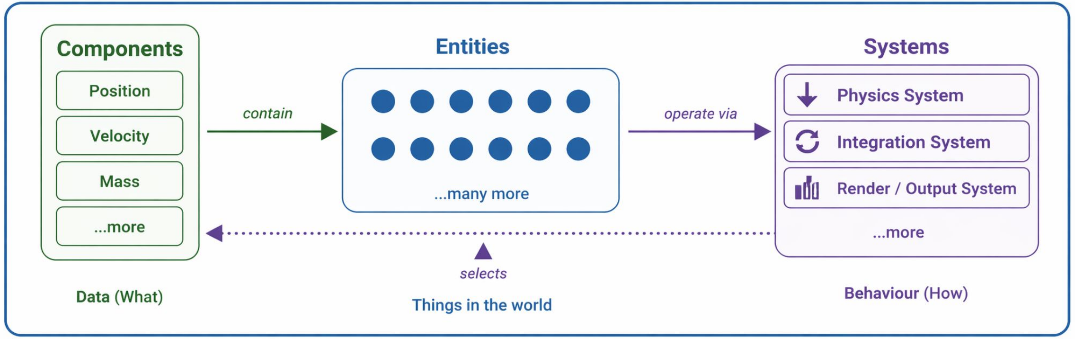
(for ECRs and others)

Entity Component Systems - ECS

- **Widely used in media** for data parallel simulation (film, SFX, games)
- **Separates data from behaviour.**
- Entities contain components.
- **Systems operate a single transform across many entities - highly parallel**
- Systems are highly focussed, helping with processor cache coherency at scale.

- Inspired by physics
- **Rarely used outside media**
- AtomECS (Cambridge) notable exception

GreenPhysECS - explores ECS utility for ECRs to build scalable research software



Constraints & Tooling

Technical Overview

- C++20 codebase
- Flecs as the ECS library
- RAPL-based energy tracking library
- Docker for consistent cross builds

Multicore & GPU deferred due to timing constraints - BUT the code is naturally concurrent.

Practical Constraints

- Project undertaken by MPhys students
- Each project was 1 semester long, approx 2 days/week
- Steep learning curve (low C++ knowledge)
- Laptop based development and simulation
  - Consequences for energy tracking testing and diversity of tested platforms

Benefits:

- Reproducible builds across Mac, Windows, Linux
  - https://github.com/UofM-Green-Compute/GreenPhysECS
- Can't get any earlier ECR than MPhys!
- Focus on practicality over perfection

The University of Manchester

The projects

Testing this hypothesis requires real work

- Fluid Dynamics
- Epidemiology - general and mitigations for outbreaks in crops

A flavour of this work follows - GreenPhysECS is extracted from this work

MANCHESTER 1824

The University of Manchester

Project 1: Fluid Dynamics

Project 1: Fluid Dynamics

- Computational Fluid Dynamics is a well developed field
- Different numerical techniques exist
  - Can explore different utility of ECS in different cases
- Two techniques were considered:
  - Finite Difference Method
  - Smoothed Particle Hydrodynamics

Two approaches: Eulerian and Lagrangian

The University of Manchester

Eulerian

- Fixed positions
- More matrix and/or differential equations based

Lagrangian

- Fixed parcels of fluid flow
- More dynamics based

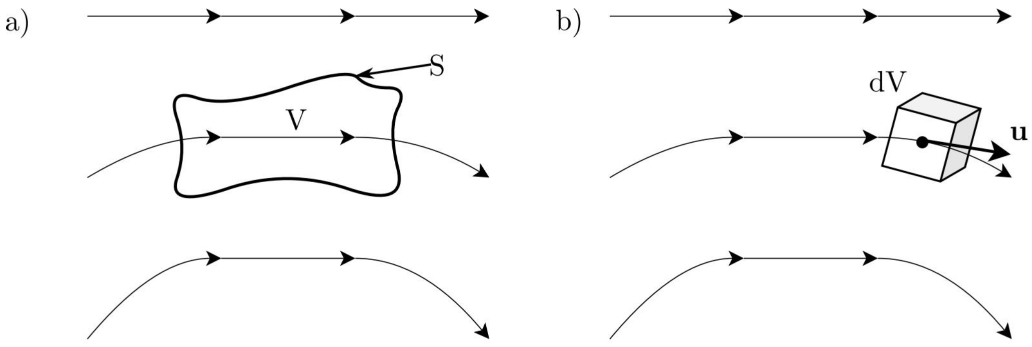


Figure 2: Diagram showing the a) the Eulerian picture of a fluid flow, and b) the Lagrangian picture of a fluid flow

Finite Difference Method (FDM)

- The FDM approximates a continuous domain by a set of discrete points
- Derivatives can then be approximated by considering the value of a function at neighbouring points

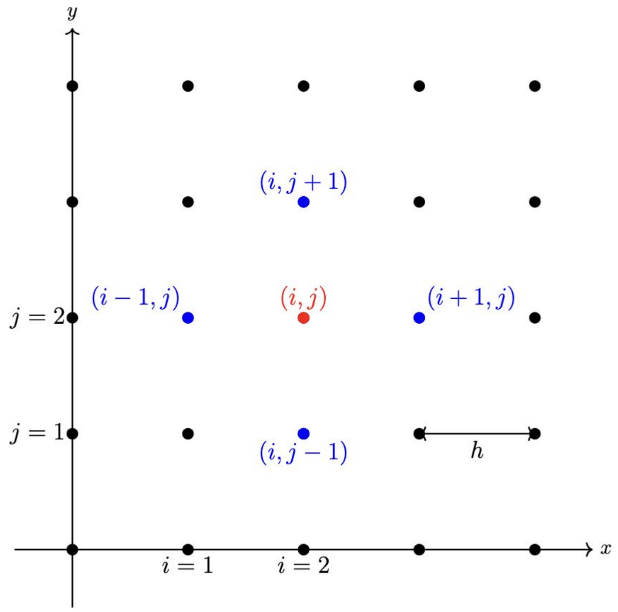


Figure 3: Figure showing a two dimensional cartesian mesh.

Smoothed Particle Hydrodynamics (SPH)

MANCHESTER 1824
The University of Manchester

- Smoothed particle hydrodynamics (SPH) is a lagrangian approach to simulating a fluid
- The approach relies on approximating the macroscopic properties of a fluid (pressure, density, etc) based on the distribution of the fluid elements
- This reduces to an equation of motion for each particle (essentially F = ma)

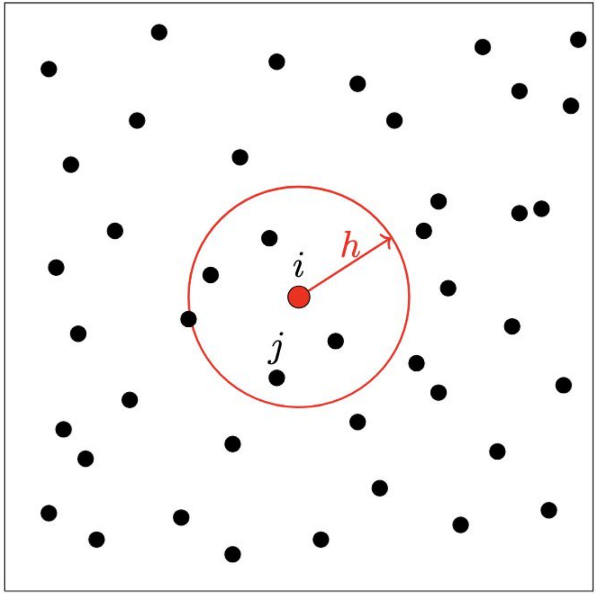


Figure 4: Figure showing SPH particles in a box. The radius around each particle represents the smoothing radius around each particle.

Smoothed Particle Hydrodynamics (SPH)

The University of Manchester

- Approximate a function based on a knows distribution of fluid elements
- Particles within some radius, h, of each other are summed over to estimate the value of a property (e.g. density)
- An interpolation function is used to weight this sum
- $$\rho(r) = \sum_b m_b W(r - r_b, h)$$

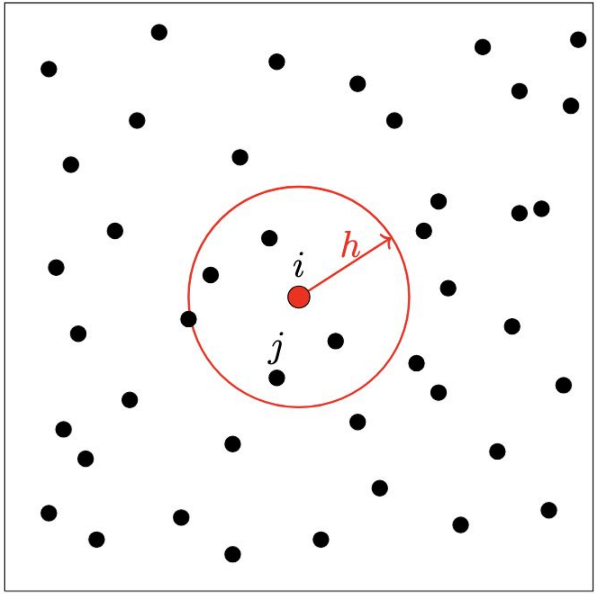


Figure 4: Figure showing SPH particles in a box. The radius around each particle represents the smoothing radius around each particle.

Implementation - Fluid in a box

The University of Manchester

FDM:

- Each point in the grid was an entity
- At each point the position, velocity, density, were components
- Systems evolve according to the discretized euler equations

SPH:

- Entities were particles of the fluid
- Components were the position, velocity, and acceleration
- Systems were used to implement collisions and evolve the components of the particle according the the equation of motion

Fluid Dynamics Results

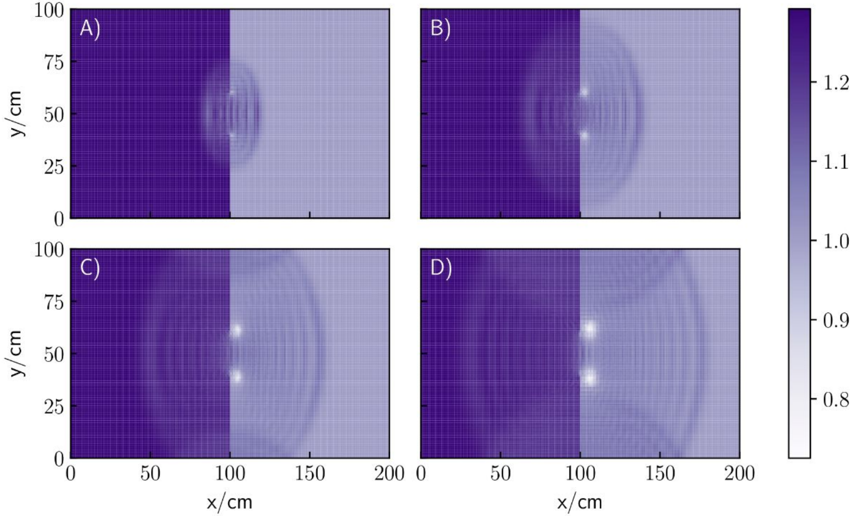


Figure 5: Graphs showing the evolution of the density plot (FDM). The graphs are evaluated at times A) t = 0.001s, B) t = 0.002, C) t = 0.003s, and D) t = 0.004s.

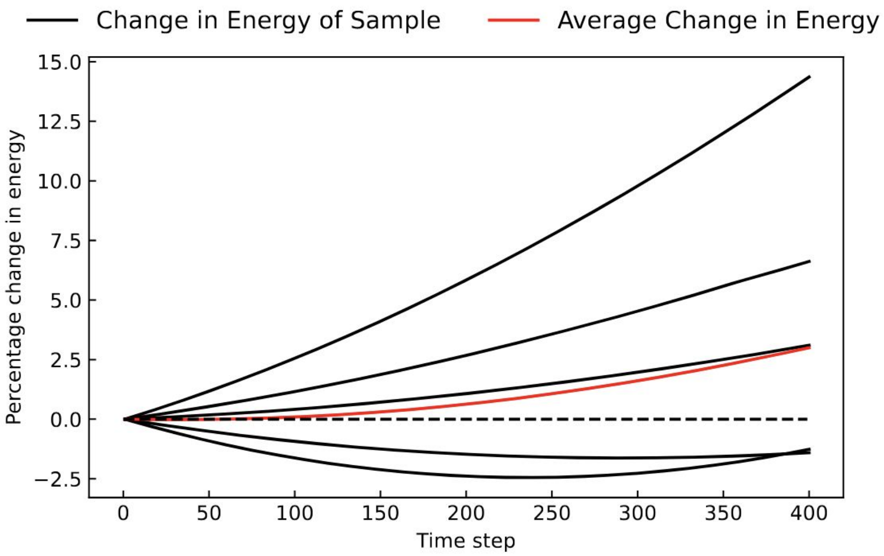


Figure 6: Change in total energy of the system for a time step t = 0.000001

Energy Tracking

- **Energy tracking was implemented** during this phase of work
- This uses an API **using RAPL** measurements and CodeCarbon but for C++
- This is implemented **as an extra SYSTEM** that runs alongside the other systems
- **This allows "book-ending" others system** for measurement
- That means the energy tracking can be dropped into any ECS based simulation easily

This serves as a good proof of principle and needs extra work

Fluid Dynamics Takeaways

Pros

- **ECS shines most where many small entities with similar properties can be created**
- **ECS also shines where the system is active and dynamic**
- **ECS could deal with time-dependent fixed-position differential equations**
    - Not entirely intuitive

Cons:

- ECS was less suited to solving matrix problems - conventional methods are the more natural choice here.

The University of Manchester

Project 2: Markov Processes

Markov Processes

- Memoryless stochastic processes
- Widely applicable
- Multiple approaches
  - Matrices
  - Calculus
  - Algorithms

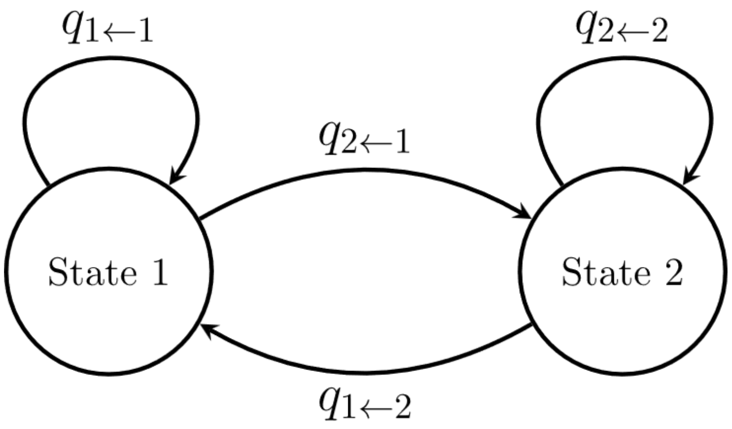


Figure 7: 2-State Markov Chain Diagram

Discrete Time Method

Transition elements:

- $q_{n\leftarrow m} = P(X(t+\Delta t)=n \,|X(t)=m)$
- Δt: time step

1. Set an initial state n=n(0)
2. Determine all possible transitions and their transition elements
3. Draw a random number between 0 and 1 and consequently choose a transition
4. Advance the simulation by Δt
5. Return to step 2

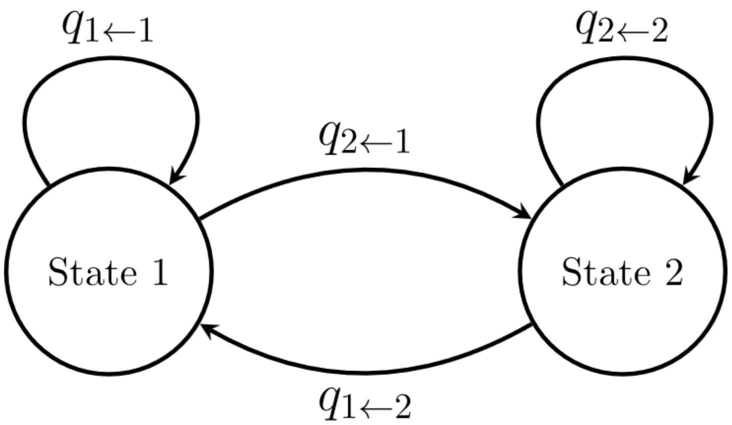


Figure 7: 2-State Markov Chain Diagram

Disease Outbreak Modelling: SIR

- Basic model: SIR model
- Each individual can be in one of three states:
  - S - Susceptible
  - I - Infected
  - R - Recovered

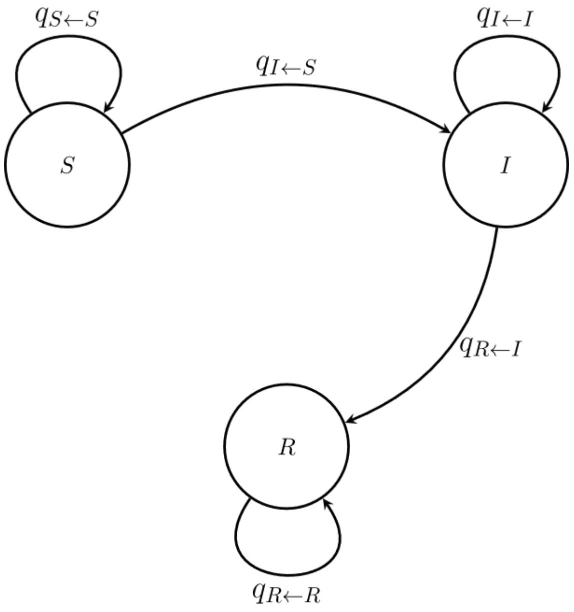


Figure 8: SIR Markov Chain Diagram

Disease Outbreak Modelling: SIR

- Transition Rates
  - $q_{s \to I} = \beta n_I \Delta t$
  - $q_{I \to R} = \gamma \Delta t$
- β: Infection rate
- γ: Recovery rate

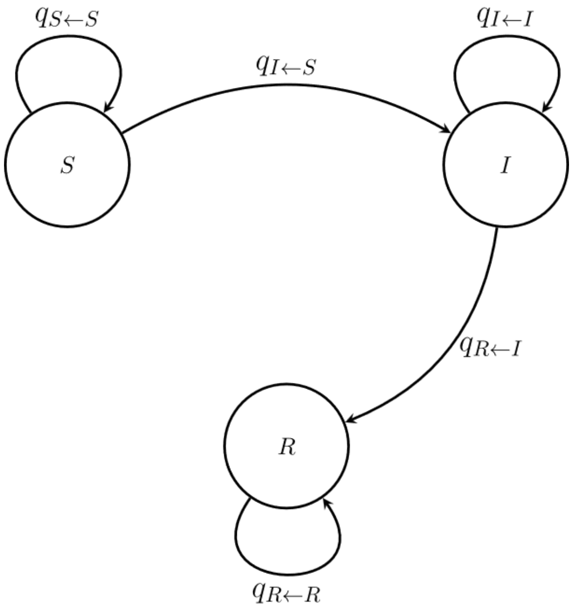


Figure 8: SIR Markov Chain Diagram

Disease Outbreak Modelling: ECS

- Entities:
  - Members of the population
- Components:
  - State
  - Index
  - Transition Probabilities
- Systems:
  - Update Transition Probabilities
  - Update State

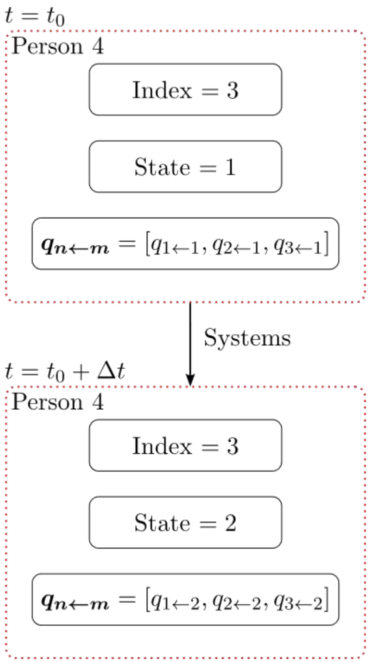


Figure 9: Systems updating the components of an entity

Disease Outbreak Modelling: SIR Results

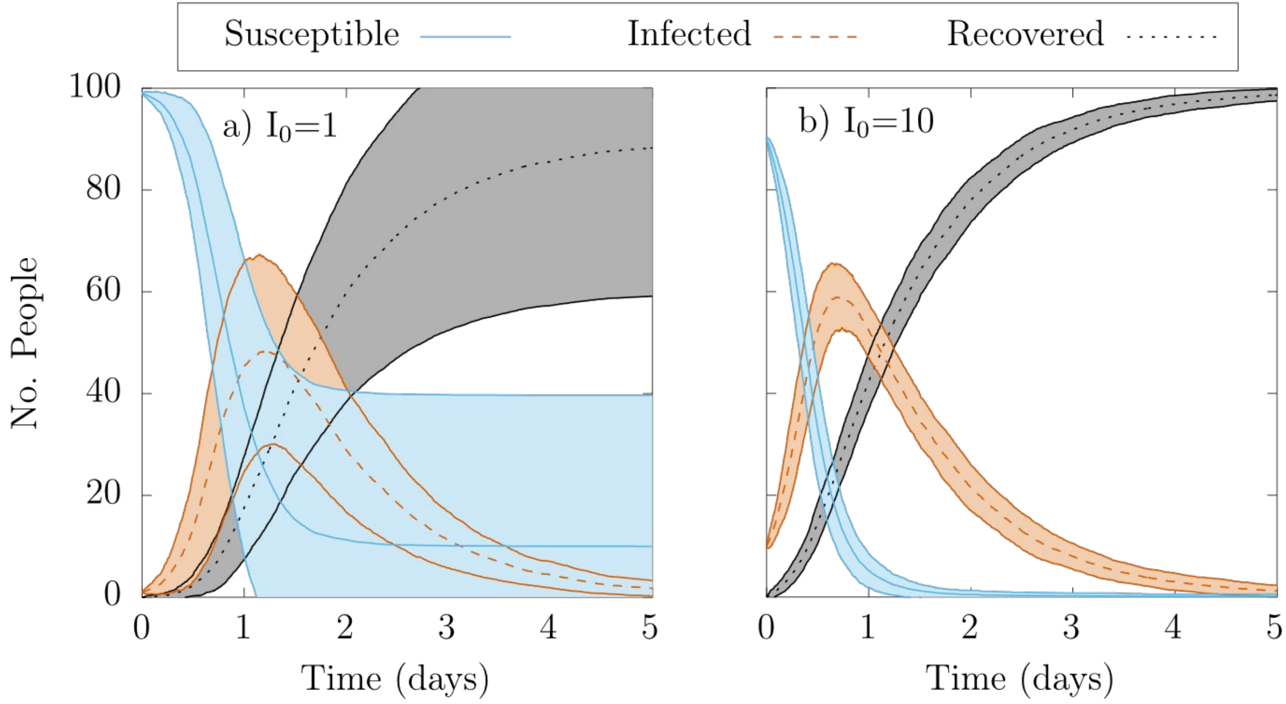


Figure 10: Ensemble averaged SIR model. □ = 7/N where N=100 is the total population γ=1. a) There is initially one infected member of the population. b) There are initially 10 infected members of the population

Disease Outbreak Modelling: Further Work

- Different Compartmental models
  - Eg SEIR
- Network Models
- Spatial Models
  - In particular a spatial Healthy-Undetectable-Detectable (HUD) model for plant epidemics

Disease Outbreak Modelling: ECS Takeaways

- ECS is well suited to epidemic modelling where many entities are involved
- Extension of existing epidemiological models is simple to implement in ECS

Conclusions

MANCHESTER 1824

The University of Manchester

Conclusions and Future

Conclusions

- Early results suggest ECS is useful for ECRs
- Code is naturally concurrent, lending itself to parallel optimisation
- Went beyond the literature surveyed - represents new results in the field

Open Questions

- Does the approach generalise?
- Which problems suit ECS best? (beyond fluid dynamics, epidemiology, etc)
- Can energy awareness become normal practice?

- Further information - https://everse.software/RSQKit/greenphsecs_research_software_story

  https://github.com/UofM-Green-Compute/GreenPhysECS

The University of Manchester

References

[1] Sander Mertens, “Flecs: A Fast Entity Component System (ECS) for C and C++.” https://www.flecs.dev/flecs/. Accessed: 12 May 2026.

[2] University of Oxford, “Ultracold Quantum Matter Lab Software.” https://coldatoms.web.ox.ac.uk/softwares. Accessed: 9 March 2026.

[3] J. Honerkamp, Statistical Physics: An Advanced Approach with Applications. Springer, 3rd ed., 2012.

[4] N. Van Kampen, Stochastic Processes in Physics and Chemistry. Elsevier, 3rd ed., 2007.

[5] T. Mikosch, Elementary Stochastic Calculus. World Scientific Publishing, 1998.

[6] O. Ibe, Markov Processes for Stochastic Modelling. Elsevier, 2nd ed., 2013.

[7] D. Stirzaker, Stochastic Processes & Models. Oxford University Press, 2005.

[8] G. Weiss, “Aspects and Applications of the Random Walk,” Journal of Statistical Physics, vol. 79, no. 1-2, pp. 497–500, 1994.

[9] F. Brauer et al., Mathematical Models in Population Biology and Epidemiology. Springer, 2010.

[10] C. Mora et al., “Over Half of Known Human Pathogenic Diseases can be Aggravated by Climate Change,” Nature Climate Change, vol. 12, no. 9, pp. 869–875, 2022.

[11] C. Carlson et al., “Climate Change Increases Cross-Species Viral Transmission Risk,” Nature, vol. 323, no. 7919, pp. 555–562, 2022.

[12] Sander Mertens, “Relationships.” https://www.flecs.dev/flecs/md_docs_2Relationships.html. Accessed: 9 May 2026.

[13] Sander Mertens, “Building Games in ECS with Entity Relationships.” https://ajmmertens.medium.com/building-games-in-ecs-with-entity-relationships-657275ba2c6c. Accessed: 9 May 2026.

[14] M. Evans, “Entity-Component-System Model: Application to Computational Fluid Dynamics Simulations and Implementation of Energy Tracking.” Unpublished MPhys report, University of Manchester, 2025.

[15] W. Leo, “1. Introduction to Probability Modeling,” in Statistical Methods for Physical Science, vol. 28 of Methods in Experimental Physics, pp. 1–34, Academic Press, 1994.

[16] J. Tolles et al., “Modeling Epidemics With Compartmental Models,” JAMA, vol. 323, no. 24, pp. 2515–2516, 2020.

The University of Manchester

References

[17] A. Lloyd, “Destabilization of Epidemic Models with the Inclusion of Realistic Distributions of Infectious Periods,” Proceedings of the Royal Society B: Biological Sciences, vol. 268, no. 1470, pp. 985–993, 2001.

[18] E. Southall et al., “A Practical Guide to Mathematical Methods for Estimating Infectious Disease Outbreak Risks,” Journal of Theoretical Biology, vol. 562, no. 111417, 2023.

[19] T. Granger et al., “Stochastic Compartment Model with Mortality and Its Application to Epidemic Spreading in Complex Networks,” Entropy, vol. 26, no. 5, article 362, 2024.

[20] M. Newman, “The Structure and Function of Complex Networks,” SIAM Review, vol. 45, no. 2, pp. 167–256, 2003.

[21] J. J. Burdon and G. A. Chilvers, “Controlled Environment Experiments on Epidemics of Barley Mildew in Different Density Host Stands,” Oecologia, vol. 26, no. 1, pp. 61–72, 1976.

[22] J. C. Douma and R. Noordhoek, “The Effect of Plant Host Density on Disease Incidence—A Meta-Analysis,” Plant, Cell & Environment, vol. 48, no. 9, pp. 6632–6644, 2025.

[23] O. Delano, “Entity-Component-System Design: Application to Computational Fluid Dynamics and Energy Cost Tracking,” Master’s thesis, University of Manchester, Manchester, England, 2025.

[24] C. W. Gardiner, Stochastic Methods: A Handbook for the Natural and Social Sciences. Springer Series in Synergetics. Berlin: Springer, 4th ed., 2009.

[25] D. T. Gillespie, Markov Processes: An Introduction for Physical Scientists. Boston: Academic Press, 1992.

[26] Food and Agriculture Organization of the United Nations, “Plant Health and the Right to Food,” 2025. https://openknowledge.fao.org/handle/20.500.14283/cd5087en. Accessed: 9 April 2026.

[27] R. A. C. Jones, “Global plant virus disease pandemics and epidemics,” Plants (Basel), vol. 10, no. 2, p. 233, 2021.

[28] J. B. Ristaino et al., “The Persistent Threat of Emerging Plant Disease Pandemics to Global Food Security,” Proceedings of the National Academy of Sciences, vol. 118, no. 23, pp. 1–9, 2021.

[29] F. A. Lovell-Read, S. Parnell, N. J. Cunniffe, and R. N. Thompson, “Using ‘Sentinel’ Plants to Improve Early Detection of Invasive Plant Pathogens,” PLoS Computational Biology, vol. 19, no. 2, 2023.

[30] P. Redmond, J. Castello, J. M. Calderón Trilla, and L. Kuper, “Exploring the Theory and Practice of Concurrency in the Entity-Component-System Pattern,” Proceedings of the ACM on Programming Languages, vol. 9, no. OOPSLA2, pp. 1–28, 2025.

[31] L. E. Reichl, A Modern Course in Statistical Physics. Weinheim, Germany: Wiley-VCH, 4th revised and updated ed., 2016

References

The University of Manchester

[32] L. E. Reichl, A Modern Course in Statistical Physics. New York: Wiley, 2nd ed., 1998.

[33] R. Bonita, R. Beaglehole, and T. Kjellström, Basic Epidemiology. Geneva: World Health Organization, 2nd ed., 2006.

[34] F. Brauer, P. van den Driessche, J. Wu, and L. J. S. Allen, Mathematical Epidemiology. Berlin: Springer, 2008.

[35] E. Kuhl, Computational Epidemiology: Data-Driven Modeling of COVID-19. Cham: Springer, 2021.

[36] K. Ayers and M. Kooiker, “A Structural Analysis of Population Graphs,” 2025.

[37] J. S. Blakemore, Solid State Physics. Cambridge: Cambridge University Press, 2nd ed., 1985.

[38] N. W. Ashcroft and N. D. Mermin, Solid State Physics. New York: Holt, Rinehart and Winston, 1976.

[39] P. K. Kundu et al., Fluid Mechanics. Academic Press, 7th ed., 2025.

[40] Clay Mathematics Institute, “Millennium Problems: The Navier–Stokes Equation.” https://www.claymath.org/millennium/navier-stokes-equation/. Accessed: 16 December 2025.

[41] R. Shen et al., “Recent application of computational fluid dynamics (cfd) in process safety and loss prevention: A review,” Journal of Loss Prevention in the Process Industries, vol. 67, p. 104252, 2020.

[42] M. Younis et al., “High-resolution regional climate–cfd integrated modelling to inform climate responsive design of northern buildings in a changing climate,” Sustainable Cities and Society, vol. 115, 2024.

[43] Sander Mertens, “Entities and Components.” https://www.flecs.dev/flecs/md_docs_2EntitiesComponents.html. Accessed: 18 December 2025.

[44] G. C. King, Vibrations and Waves. Wiley, 2009.

[45] X. Wang et al., Computational Methods for Nonlinear Dynamical Systems. Elsevier, 2023.

[46] S. Mazumder, Numerical Methods for Partial Differential Equations. Academic Press, 2016.

[47] J. Butcher, “A History of Runge–Kutta Methods,” Applied Numerical Mathematics, vol. 20, no. 3, pp. 247–260, 1996.

MANCHESTER 1824
The University of Manchester

References

[48] J. Blazek, Computational Fluid Dynamics: Principles and Applications. Butterworth-Heinemann, 3rd ed., 2015.

[49] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics. Springer, 3rd ed., 2009.

[50] Engineering ToolBox, “Air—Thermal Diffusivity vs. Temperature and Pressure.” https://www.engineeringtoolbox.com/air-thermal-diffusivity-d_2011.html. Accessed: 18 December 2025.

[51] Engineering ToolBox, “Air Density, Specific Weight, and Thermal Expansion Coefficients at Varying Temperatures and Pressures.” https://www.engineeringtoolbox.com/air-density-specific-weight-d_600.html. Accessed: 18 December 2025.

[52] N. Balakrishnan and G. Fernandez, “Wall Boundary Conditions for Inviscid Compressible Flows on Unstructured Meshes,” International Journal for Numerical Methods in Fluids, vol. 28, pp. 1481–1501, 1998.

[53] G. A. Sod, “A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws,” Journal of Computational Physics, vol. 27, no. 1, pp. 1–31, 1978.

[54] X. Chen, M. Zeuner, U. Schneider, C. Foot, T. Harte, and E. Bentine, “AtomECS: Simulate Laser Cooling and Magneto-Optical Traps,” arXiv preprint arXiv:2105.06447, 2021.

[55] J. H. Ferziger, M. Perić, and R. L. Street, Computational Methods for Fluid Dynamics. Cham, Switzerland: Springer, 4th ed., 2020.

[56] F. Mandl, Statistical Physics. Manchester Physics Series, Hoboken: Wiley, 2nd ed., 2013.

[57] R. Gingold and J. Monaghan, “Kernel Estimates as a Basis for General Particle Methods in Hydrodynamics,” Journal of Computational Physics, vol. 46, no. 3, pp. 429–453, 1982.

[58] R. A. Gingold and J. J. Monaghan, “Smoothed particle hydrodynamics: theory and application to non-spherical stars,” Monthly Notices of the Royal Astronomical Society, vol. 181, no. 3, pp. 375–389, 1977.

[59] A. G. Barbu and S. C. Zhu, Monte Carlo Methods. Singapore: Springer, 2020.

BACKUP Slides

Backup Slides

ECS Context

- Entity-Component-System (ECS) is a software development pattern that encourages the separation of data (components) from behaviour (systems)
- ECS programs are naturally concurrent, modular, and flexible

What is ECS

- Entities: The physical or logical object we are trying to simulate, **e.g. a particle**
- Components: Data associated with an entity, **e.g. the particle's velocity**
- Systems: Functions to alter the components of entities that match some criteria, **e.g. a collision system**

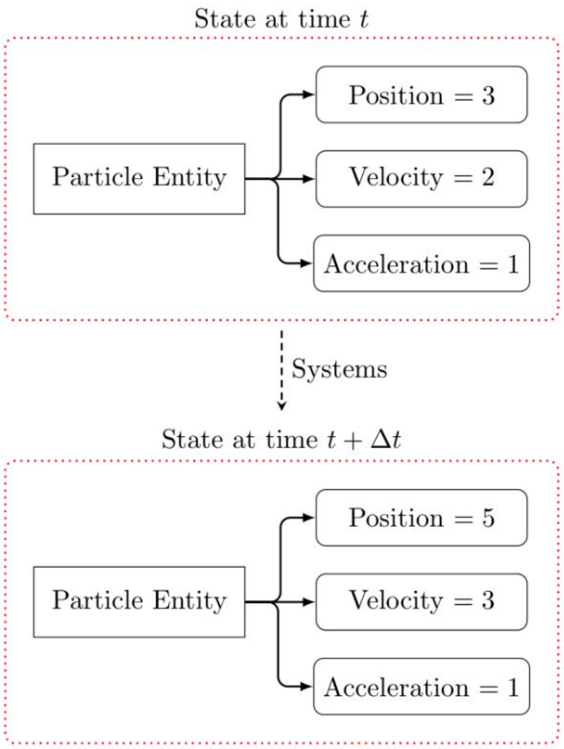


Figure 1: Diagram showing the evolution of the state of an ECS program. The state of the simulation is characterised by the entities and components. Systems evolve this state.

Modelling HUD Crop Epidemiology on a Lattice

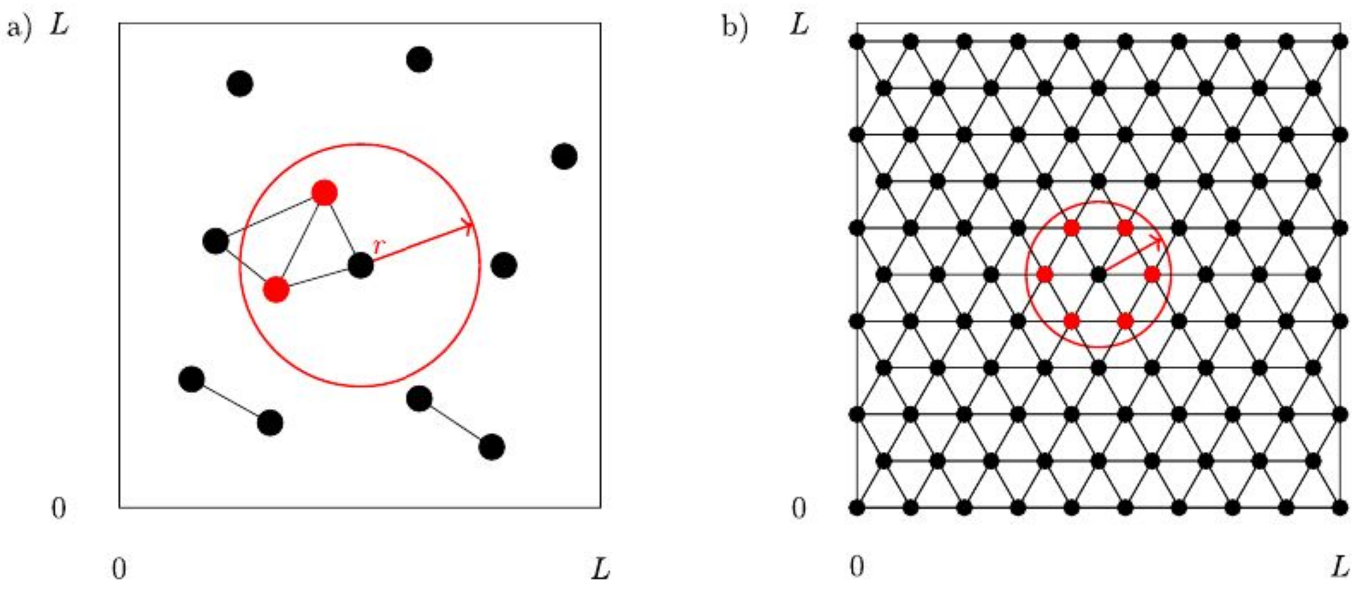


Figure 11: a) Plants are randomly placed in an $L \times L$ area. Plants within some radius, R, of each other are linked on a network. Linked plants can spread the infection to one another. b) Plants are placed in a hexagonal lattice in a $L \times L$ area. Similarly, all plants within R of each other are linked on a network.

Modelling HUD Crop Epidemiology on a Lattice

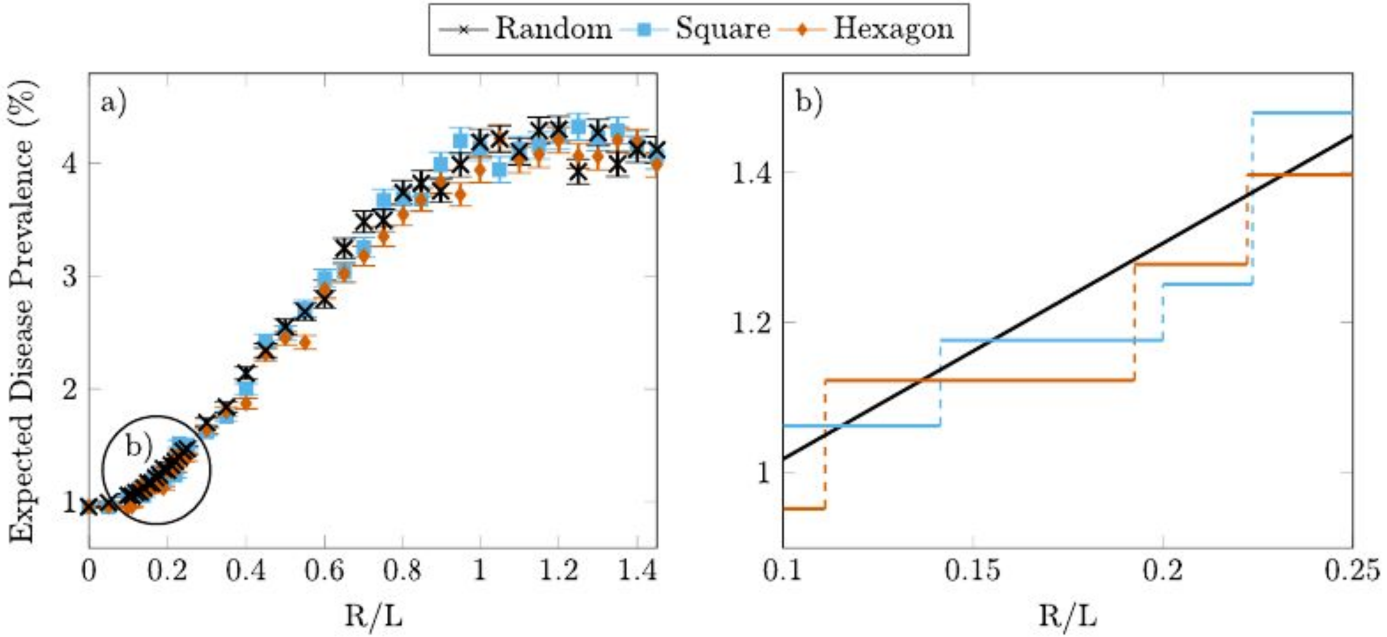


Figure 12: a) The *EDP* for infection radii between 0 and 1.45L, where L is the box length. b) A closer look at the *EDP* for infection radii between 0.1L and 0.25L with particular emphasis on the plateau behaviour of the lattice cases.

Early Detection of Disease Outbreak in Crops - HUD Model