Making squishy maps

Big-O London

9 Feb 2015

Robin Houston

KILN

Data + journalism + technology + design = KILN

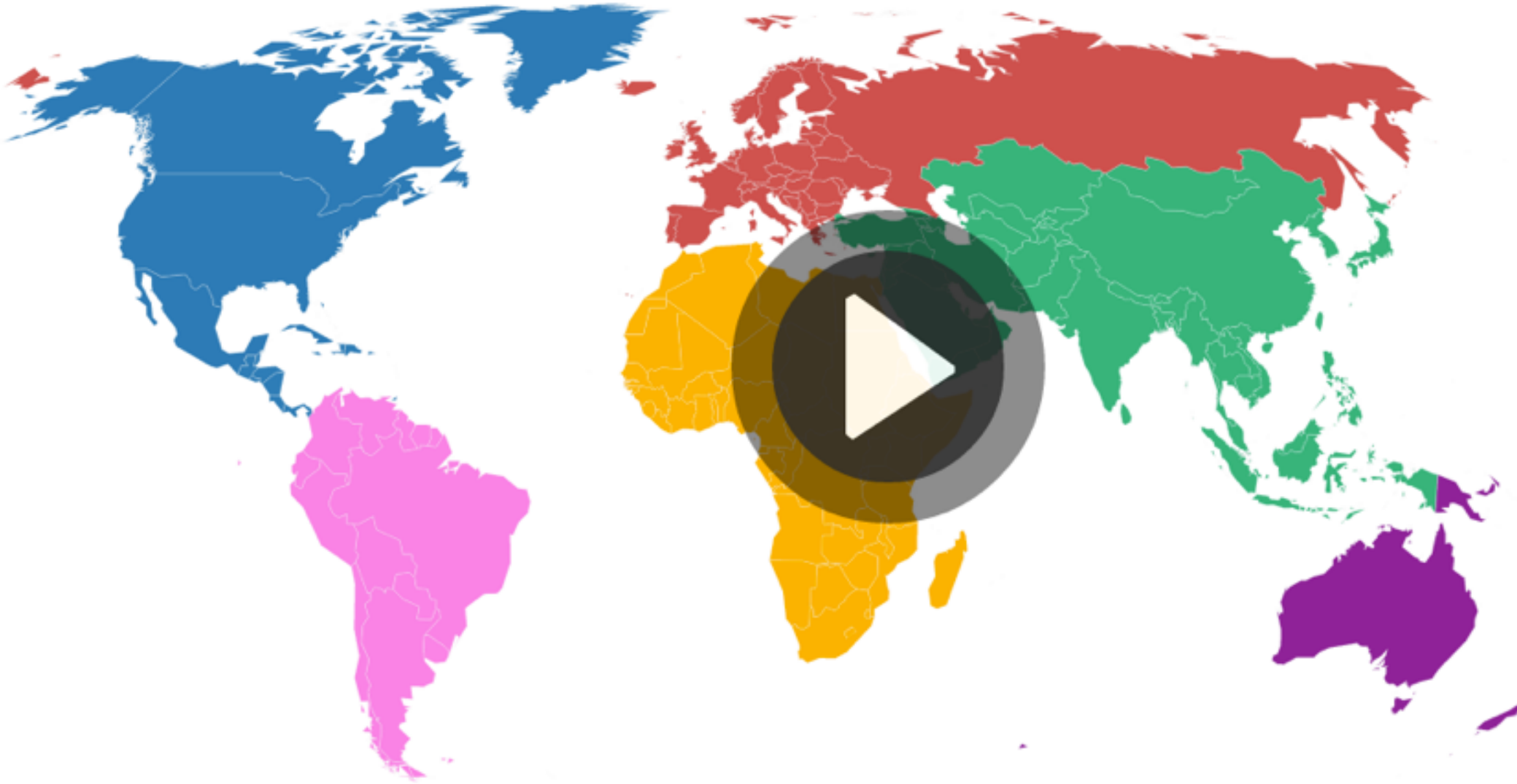
Kiln produces interactives, maps and visualisations that bring complex data and stories to life. We also provide data visualising training.

The Carbon Map

BACKGROUND
Area Population Wealth

RESPONSIBILITY
Extraction Emissions Consumption Historical Reserves

VULNERABILITY
People at risk Sea level Poverty



Shade by
Continents
Europe
Africa
Asia
North America
South America
Oceania

This map

This is the world as seen in one of the most common maps: the Robinson projection.

ABOUT THIS TOOL DATA SOURCES

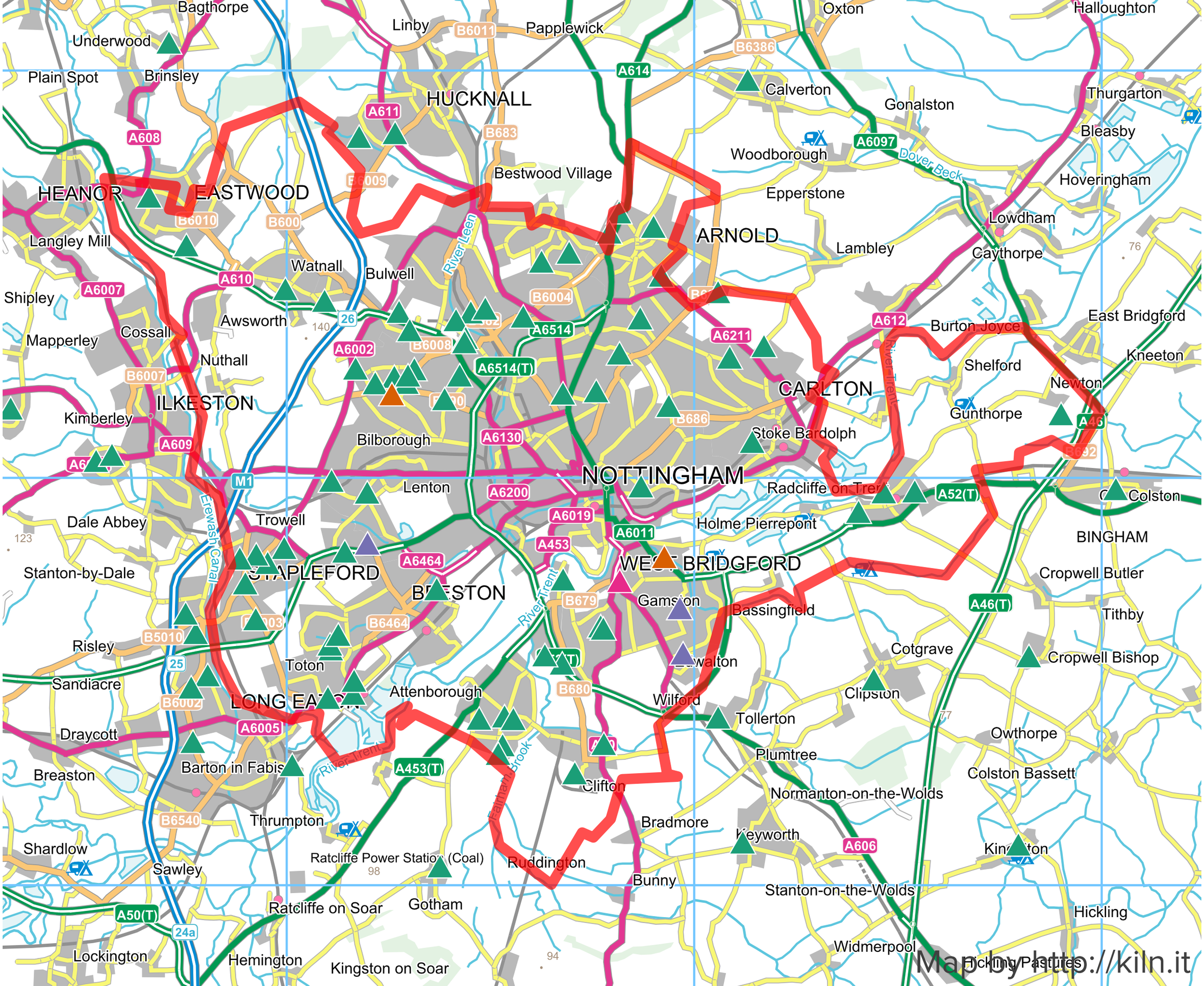
Continuous area cartograms

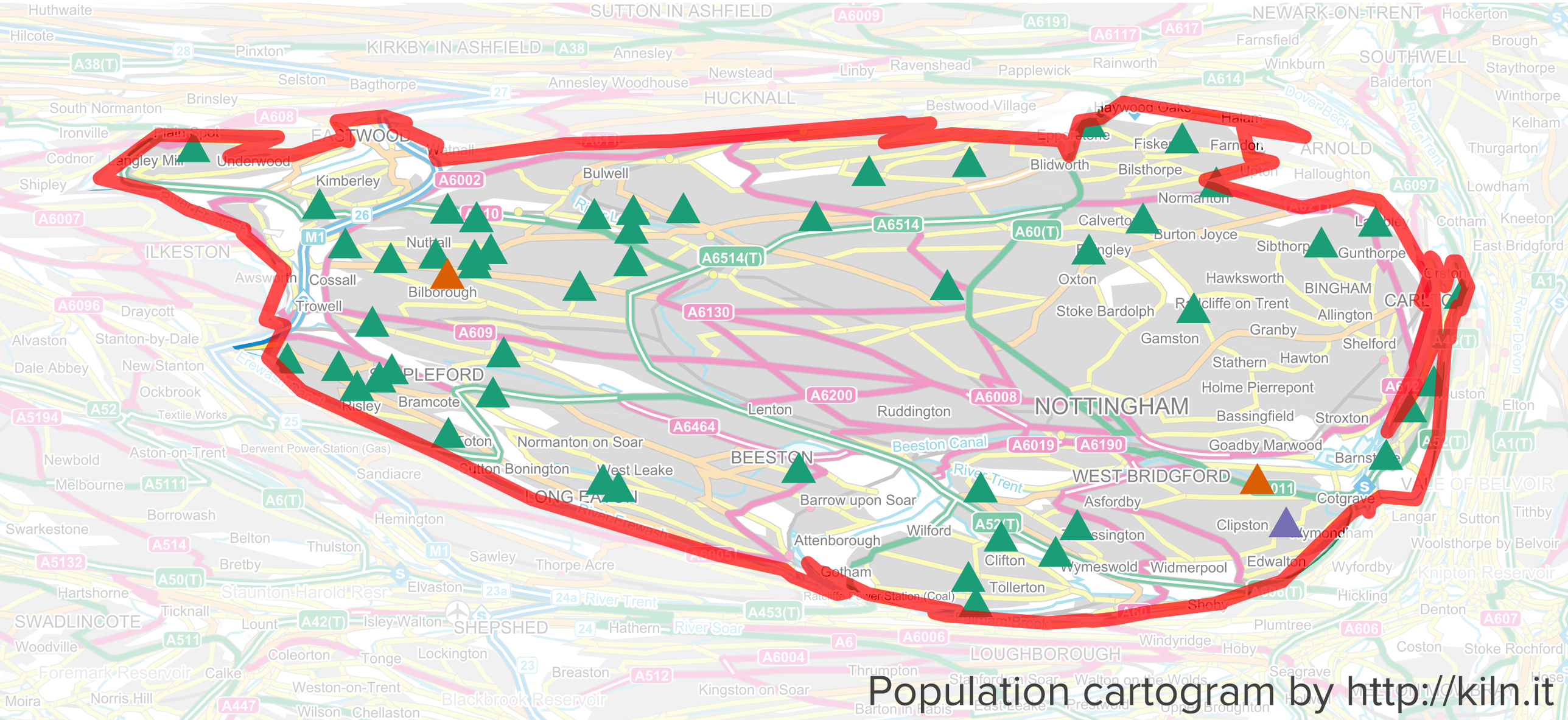
- Distort the shapes of countries (or other regions) to represent some variable of interest
- There are some isolated early examples made by hand, but it really began in the 1970s as computers started to be used for cartography
- The earliest algorithm I know of is by cartographic pioneer Wadlo Tobler (1973)
- The method of Dougenik et al. (1984) is still used
- But I'm going to talk about my favourite: the diffusion algorithm of Michael Gaster and Mark Newman (2004)

KILN

Human population

Mark Newman, 2009


Bagthorpe
Underwood
Linby
B6011
Papplewick
Oxton
Hallloughton
Plain Spot
Brinsley
A614
B6386
Calverton
Thurgarton
HUCKNALL
A611
Gonalston
A608
B683
Bleasby
A6097
Dover Beck
Woodborough
Bestwood Village
B6009
Hoveringham
HEANOR
EASTWOOD
Epperstone
B6010
B600
Lowdham
ARNOLD
Langley Mill
Lambley
Caythorpe
River Leen
Watnall
Bulwell
A610
B6004
Shipley
A6007
Awsworth
A612
Burton Joyce
East Bridgford
Cossall
A6514
A6211
Mapperley
A6002
B6008
Kneeton
Nuthall
A6514(T)
Shelford
River Trent
Newton
B6007
CARLTON
ILKESTON
Gunthorpe
Kimberley
B686
Stoke Bardolph
A609
Bilborough
A6130
A46
B692
NOTTINGHAM
M1
Lenton
A6200
Radcliffe on Trent
A52(T)
Colston
Dale Abbey
Trowell
A6019
Holme Pierrepont
A6011
BINGHAM
Erewash Canal
A453
Stanton-by-Dale
STAPLEFORD
A6464
WEST BRIDGFORD
Cropwell Butler
BEESTON
River Trent
B679
Gamston
A46(T)
Bassingfield
Tithby
B5010
B6464
Risley
Cotgrave
Cropwell Bishop
Toton
Sandiacre
Attenborough
B680
Wilford
Clipston
B6002
LONG EATON
Tollerton
A6005
Draycott
Owthorpe
Barton in Fabis
River Trent
Plumtree
A453(T)
Breaston
Colston Bassett
Clifton
Normanton-on-the-Wolds
B6540
Thrumpton
Bradmore
Keyworth
Shardlow
A606
Ratcliffe Power Station (Coal)
Ruddington
Sawley
Bunny
Stanton-on-the-Wolds
A50(T)
Ratcliffe on Soar
Gotham
Hickling
24a
Widmerpool
Lockington
Hemington
Kingston on Soar
Hickling Pastures
Map by http://kiln.it



Population cartogram by http://kiln.it

Michael Gastner

Mark Newman

How does it work?

- Numerical simulation of the process of diffusion!

KILN

Diffusion-based method for producing density-equalizing maps

Michael T. Gastner and M. E. J. Newman*

Center for the Study of Complex Systems and Department of Physics, University of Michigan, Ann Arbor, MI 48109

Edited by Michael F. Goodchild, University of California, Santa Barbara, CA, and approved April 2, 2004 (received for review January 13, 2004)

Map makers have for many years searched for a way to construct cartograms, maps in which the sizes of geographic regions such as countries or provinces appear in proportion to their population or some other analogous property. Such maps are invaluable for the representation of census results, election returns, disease incidence, and many other kinds of human data. Unfortunately, to scale regions and still have them fit together, one is normally forced to distort the regions' shapes, potentially resulting in maps that are difficult to read. Many methods for making cartograms have been proposed, some of them are extremely complex, but all suffer either from this lack of readability or from other pathologies, like overlapping regions or strong dependence on the choice of coordinate axes. Here, we present a technique based on ideas borrowed from elementary physics that suffers none of these drawbacks. Our method is conceptually simple and produces useful, elegant, and easily readable maps. We illustrate the method with applications to the results of the 2000 U.S. presidential election, lung cancer cases in the State of New York, and the geographical distribution of stories appearing in the news.

Suppose we wish to represent on a map some data concerning, to take the most common example, the human population. For instance, we might wish to show votes in an election, incidence of a disease, number of cars, televisions, or phones in use, numbers of people falling in one group or another of the population, by age or income, or any of very many other variables of statistical, medical, or demographic interest. The typical course under such circumstances would be to choose one of the standard projections for the area of interest and plot the data on it with some color code or similar representation. Such maps, however, can be misleading. A plot of disease incidence, for example, will inevitably show high incidence in cities and low

this kind are known as value-by-area maps, density-equalizing maps, or cartograms.

The construction of cartograms is a challenging undertaking. A variety of methods have been put forward, but none is entirely satisfactory. In particular, many of these methods produce highly distorted maps that are difficult to read or projections that are badly behaved under some circumstances, with overlapping regions or strong dependence on coordinate axes. In many cases the methods proposed are also computationally demanding, sometimes taking hours to produce a single map. In this article we propose a method that is, we believe, intuitive, but also produces elegant, well behaved, and useful cartograms, whose calculation makes relatively low demands on our computational resources.

Previous Methods for Constructing Cartograms

Mathematically, the construction of a (flat 2D) cartogram involves finding a transformation $\mathbf{r} \to \mathbf{T}(\mathbf{r})$ of a plane to another plane such that the Jacobian $\partial(T_x,T_y)/\partial(x,y)$ of the transformation is proportional to some specified (population) density $\rho(r)$, thus:

$$\frac{\partial(T_x, T_y)}{\partial(x, y)} \equiv \frac{\partial T_x}{\partial x}\frac{\partial T_y}{\partial y} - \frac{\partial T_x}{\partial y}\frac{\partial T_y}{\partial x} = \frac{\rho(\mathbf{r})}{\bar{\rho}}, \qquad [1]$$

where $\bar{\rho}$ is the mean population density averaged over the area to be mapped. (This choice of normalization for the Jacobian ensures that the total area before and after the transformation is the same.)

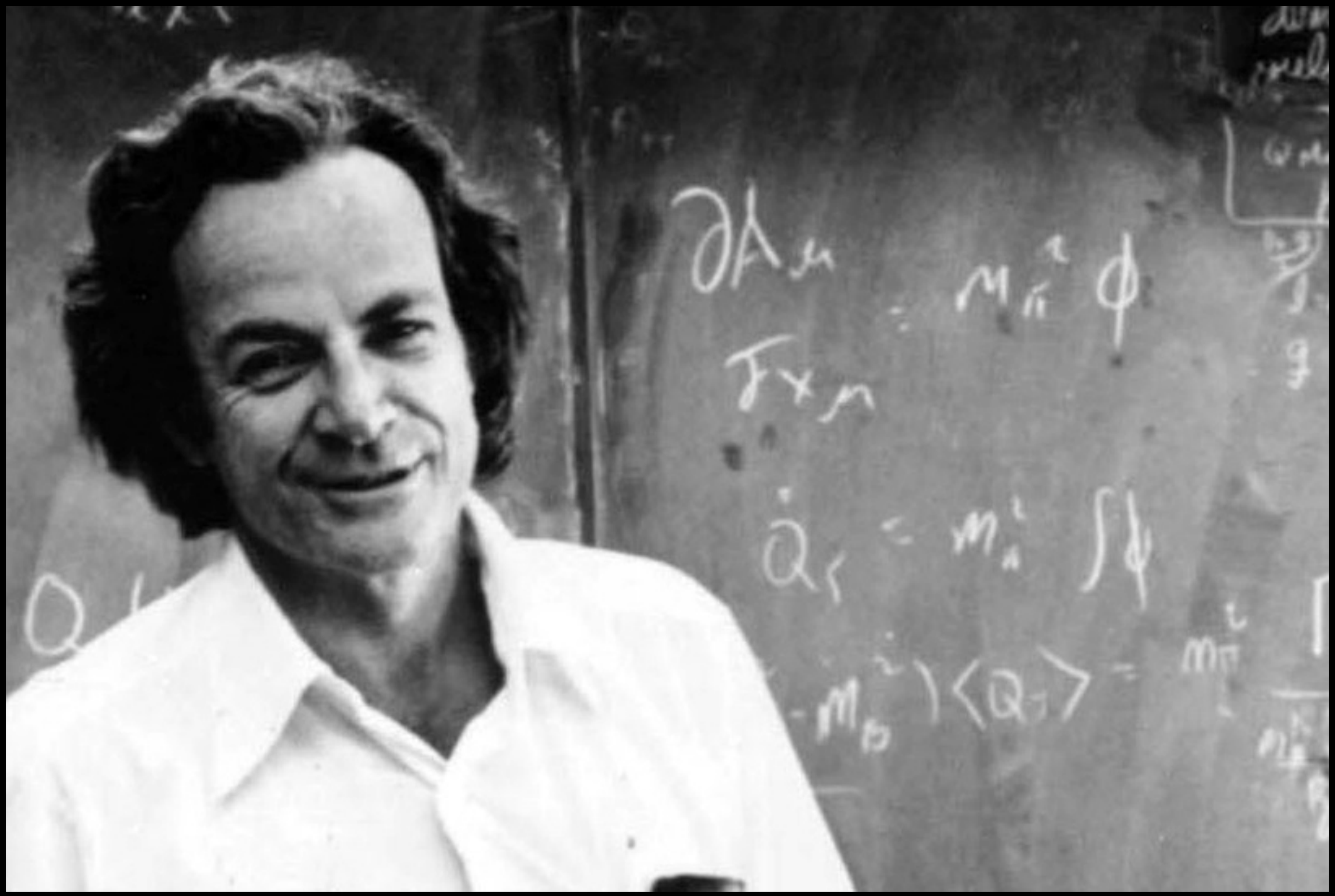
Eq. **1** does not determine the cartogram projection uniquely. To do that, we need one more constraint; two constraints are needed to fix the projection for a 2D cartogram. Different

A physicist's algorithm

- I love this algorithm partly because I found it so surprising.
- It's *very* different from anything a computer scientist would have come up with.

- What do physicists love best of all?
- Partial differential equations!

KILN



Richard Feynman and The Connection Machine

By the end of that summer of 1983, Richard had completed his analysis of the behavior of the router, and much to our surprise and amusement, he presented his answer in the form of a set of partial differential equations. To a physicist this may seem natural, but to a computer designer, treating a set of boolean circuits as a continuous, differentiable system is a bit strange. … Our discrete analysis said we needed seven buffers per chip; Feynman's equations suggested that we only needed five. We decided to play it safe and ignore Feynman.

The decision to ignore Feynman's analysis was made in September, but by next spring we were up against a wall. The chips that we had designed were slightly too big to manufacture and the only way to solve the problem was to cut the number of buffers per chip back to five. Since Feynman's equations claimed we could do this safely, his unconventional methods of analysis started looking better and better to us. We decided to go ahead and make the chips with the smaller number of buffers.

Fortunately, he was right. When we put together the chips the machine worked.

— W. Daniel Hillis, 1989

Diffusion

Start with:

Initial density: $\rho_0(x, y)$

Deduce:

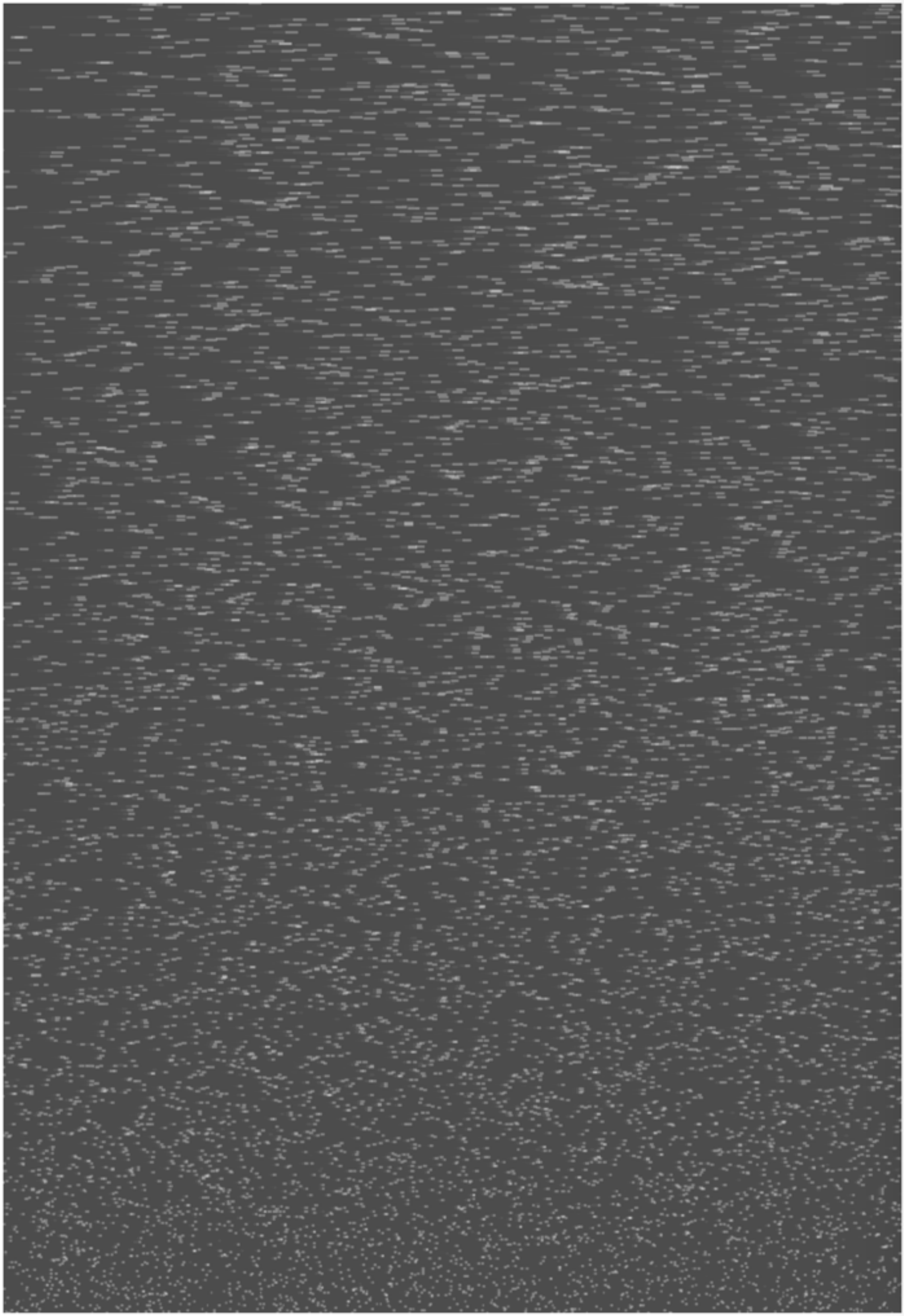
Evolution of density: $\rho(\binom{x}{y}, t)$

Evolution of velocity: $\mathbf{v}(\binom{x}{y}, t)$

Particle displacement: $\mathbf{d}(\binom{x}{y}, t)$

Desired result:

Eventual displacement: $\lim_{t \to \infty} \mathbf{d}(\binom{x}{y}, t)$



Diffusion

Initial conditions:

$$\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, 0\right) = \rho_0(x, y) \qquad \mathbf{d}\left(\begin{pmatrix}x\\y\end{pmatrix}, 0\right) = \begin{pmatrix}x\\y\end{pmatrix}$$

Relationship between quantities:

Define flux: $\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)\ \mathbf{v}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$

Continuity equation: $\frac{\partial}{\partial t}\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla \cdot \mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$

Fick's first law: $\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$

Displacement: $\frac{\partial}{\partial t}\mathbf{d}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \mathbf{v}\left(\mathbf{d}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right), t\right)$

Diffusion

$$\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)\ \mathbf{v}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\frac{\partial}{\partial t}\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla\cdot\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla\,\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\frac{\partial}{\partial t}\,\mathbf{d}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \ \mathbf{v}\left(\mathbf{d}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right), t\right)$$

Diffusion

$$\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)\ \mathbf{v}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\frac{\partial}{\partial t}\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla\cdot\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla\,\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\frac{\partial}{\partial t}\,\mathbf{d}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \mathbf{v}\left(\mathbf{d}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right), t\right)$$

*If we can compute the velocity at arbitrary time and position, we can solve for **d** numerically using standard techniques.*

Diffusion

$$\mathbf{j}\left(\binom{x}{y}, t\right) = \rho\left(\binom{x}{y}, t\right)\, \mathbf{v}\left(\binom{x}{y}, t\right)$$

$$\frac{\partial}{\partial t}\rho\left(\binom{x}{y}, t\right) = -\nabla \cdot \mathbf{j}\left(\binom{x}{y}, t\right)$$

$$\mathbf{j}\left(\binom{x}{y}, t\right) = -\nabla\, \rho\left(\binom{x}{y}, t\right)$$

Diffusion

$$\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)\ \mathbf{v}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\frac{\partial}{\partial t}\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla\cdot\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla\,\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\mathbf{v}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \frac{-\nabla\ \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)}{\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)}$$

Diffusion

$$\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)\, \mathbf{v}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\frac{\partial}{\partial t}\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla \cdot \mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\mathbf{v}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \frac{-\nabla\, \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)}{\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)}$$

$$\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla\, \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

Diffusion

$$\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)\, \mathbf{v}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\frac{\partial}{\partial t}\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla\cdot \mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\mathbf{v}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \frac{-\nabla\, \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)}{\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)}$$

$$\mathbf{j}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = -\nabla\, \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

$$\Rightarrow\ \frac{\partial}{\partial t}\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \nabla^2\, \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

Diffusion

$$\mathbf{v}\left(\binom{x}{y}, t\right) = \frac{-\nabla \rho\left(\binom{x}{y}, t\right)}{\rho\left(\binom{x}{y}, t\right)}$$

$$\frac{\partial}{\partial t}\rho\left(\binom{x}{y}, t\right) = \nabla^2 \rho\left(\binom{x}{y}, t\right)$$

Diffusion

If we know the density

at arbitrary time and position,

we can compute the density gradient

and hence the velocity.

$$\mathbf{v}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \frac{-\nabla \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)}{\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)}$$

$$\frac{\partial}{\partial t}\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \nabla^2 \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

Diffusion

$$\frac{\partial}{\partial t}\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \nabla^2 \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

Diffusion

All that remains is to solve this:

$$\frac{\partial}{\partial t}\rho\left(\binom{x}{y}, t\right) = \nabla^2 \rho\left(\binom{x}{y}, t\right)$$

Commonly known as the **Heat Equation**.

The heat equation

$$\frac{\partial}{\partial t}\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \nabla^2 \rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right)$$

To make a long story short, the solution is:

$$\rho\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \int\int \frac{1}{4\pi t}\, e^{\frac{-(x-x')^2-(y-y')^2}{4t}}\, \rho_0(x', y')\, dx'\, dy'$$

This is a convolution with a Gaussian kernel:

equivalent to a Gaussian blur of the initial density field with radius $\sqrt{2t}$

We could just integrate it numerically if we had to,

but it's more efficient to use a Fourier transform.

The Gastner-Newman algorithm

Numerically solve

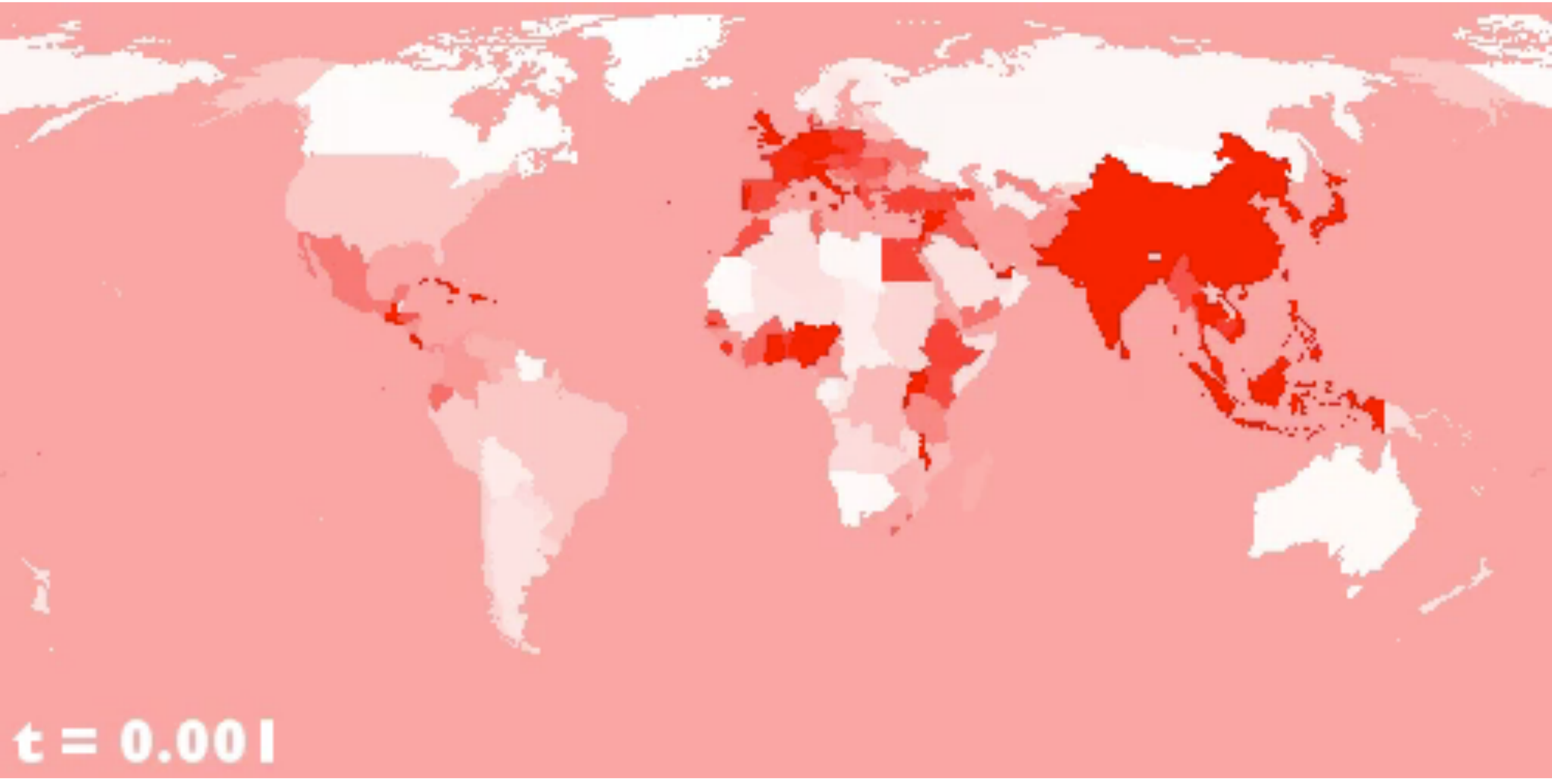
$$\frac{\partial}{\partial t}\,\mathbf{d}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right) = \mathbf{v}\left(\mathbf{d}\left(\begin{pmatrix}x\\y\end{pmatrix}, t\right), t\right)$$

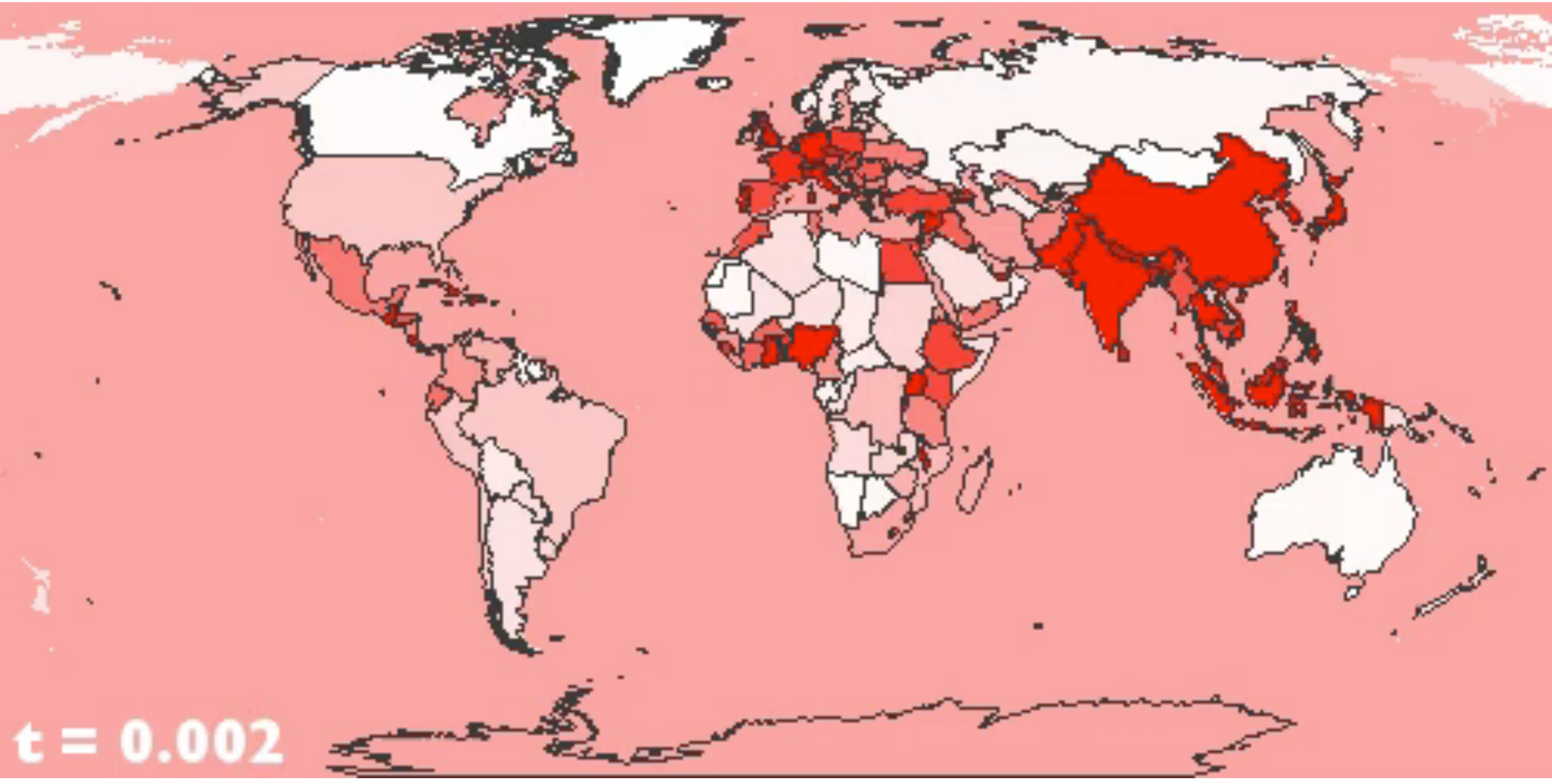
using the fourth-order Runge-Kutta method.

At each time step compute ρ using a Fourier convolution and use that to compute $\mathbf{v}$.

- You can download a pretty good implementation in C from Mark Newman's website.

KILN


t = 0.00


t = 0.002

Custom travel maps

1 Select where you've been

2 Sign in to theguardian

3 See your map, play and share

SORT BY **POPULAR** CONTINENT LETTER

Make my map! ✓

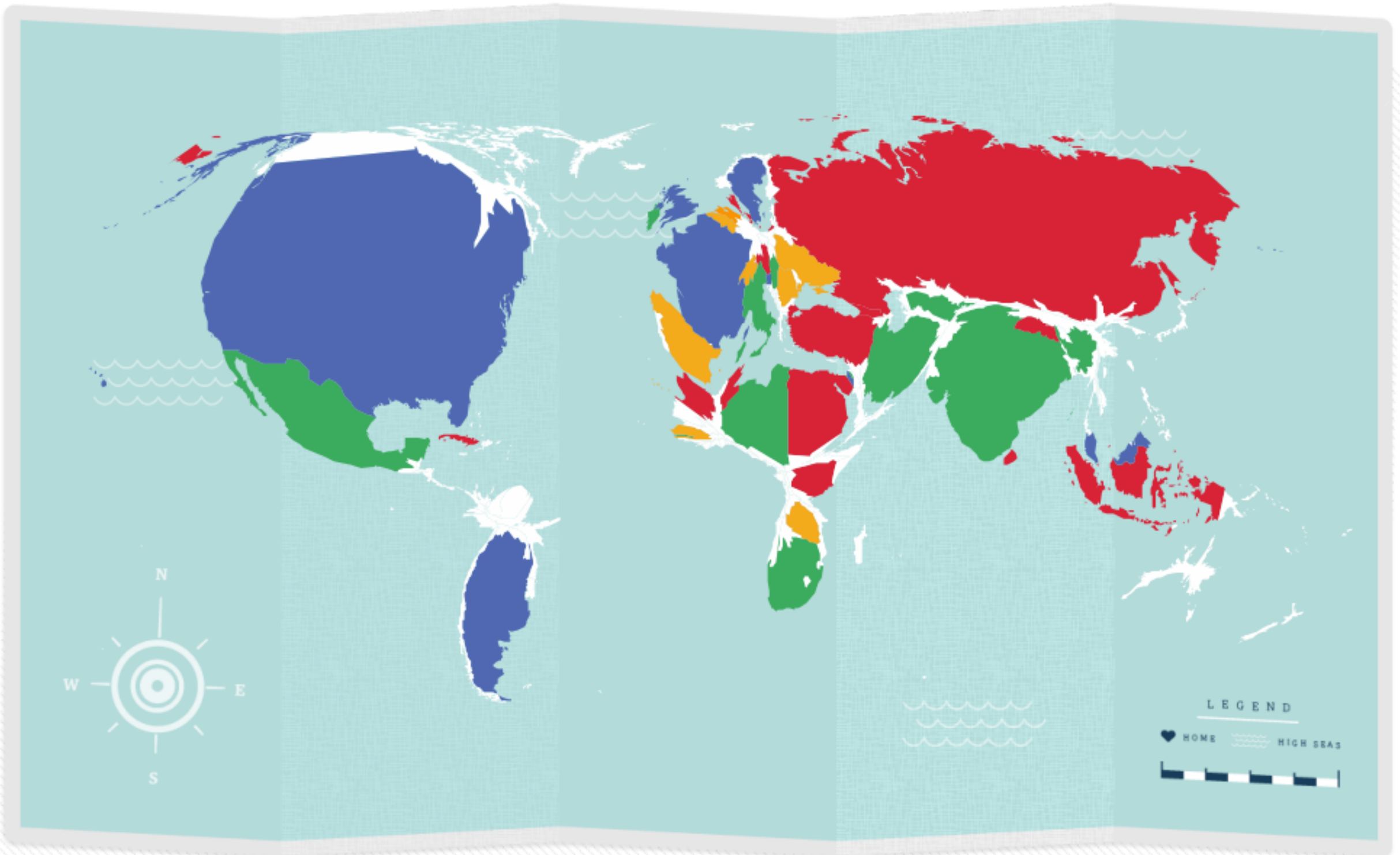
Spain

− 0 visits +

Cheating with the Central Limit Theorem

- The Central Limit Theorem implies that a Gaussian convolution can be approximated by repeatedly applying some other convolution.
- If you don't have access to a highly-optimised FFT library, the Gaussian convolution is rather expensive to compute.
- So you can cheat by using a cheap "box" filter iteratively.
- I can't prove any rigorous bounds on the cumulative error with this method, but it works perfectly well in practice.
- As well as being faster, it reduces the space requirements.

KILN

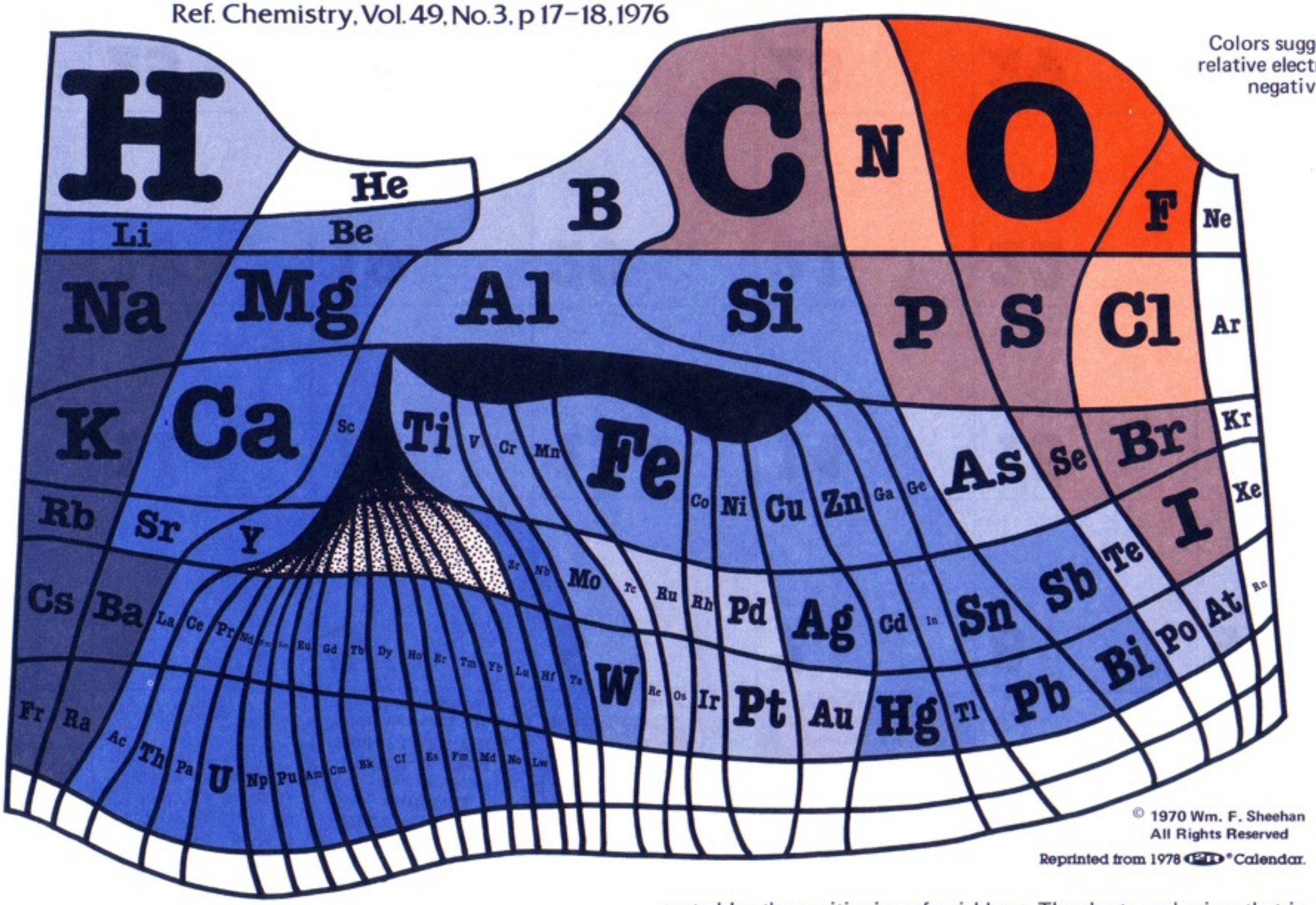

N
W
E
S
LEGEND
HOME
HIGH SEAS

The Elements According to Relative Abundance

A Periodic Chart by Prof. Wm. F. Sheehan, University of Santa Clara, CA 95053

Ref. Chemistry, Vol. 49, No. 3, p 17–18, 1976

Colors suggest relative electro-negativity



Reprinted from 1978 °Calendar.

Roughly, the size of an element's own niche ("I almost wrote square") is proportioned to its abundance on Earth's surface, and in addition, certain chemical similarities (e.g., Be and Al, or B and Si) are suggested by the positioning of neighbors. The chart emphasizes that in real life a chemist will probably meet O, Si, Al, . . . and that he better do something about it. Periodic tables based upon elemental abundance would, of course, vary from planet to planet. . . W.F.S.

NOTE: TO ACCOMMODATE ALL ELEMENTS SOME DISTORTIONS WERE NECESSARY, FOR EXAMPLE SOME ELEMENTS DO NOT OCCUR NATURALLY.

Thanks

TRAINING

Kiln regularly offers public courses in data visualization and interactive design, from introductory courses for beginners to advanced coding masterclasses for developers. The next dates are:

14 March 2015: Introduction to interactive data visualization

Details and booking

15 March 2015: Data visualization with D3

Details and booking

“A two day course of pure excellence.”

ATTENDEE AT RECENT GUARDIAN MASTERCLASS

www.kiln.it

@k i l n

hello@kiln.it