# Working with Partial Differential Equations in Matlab

Jim Ramsay, McGill University

# Overview

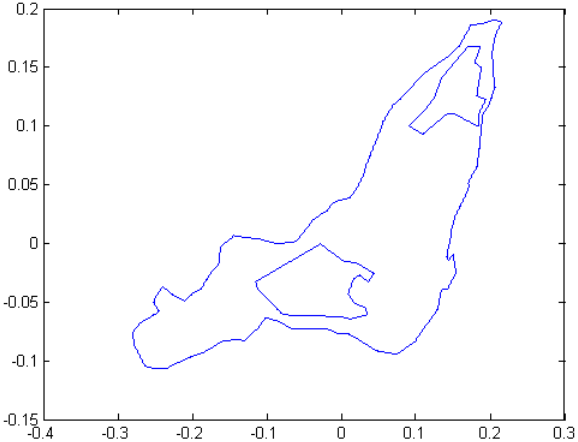
- The problem to be solved: space-time function $u$
- Two partial differentiation operators
- The equations solved by Matlab's pde toolbox
- The boundary behavior of $u$
- Using Matlab's PDE toolbox

This talk follows closely the first chapter of J. O. Ramsay and B. W. Silverman, (2005) *Functional Data Analysis, Second Edition*. New York: Springer.

# Outline

- We want to compute a function $u(x, y; t)$ that is a function of spatial coordinates $x$ and $y$ and also, possibly, of time $t$.
- The spatial coordinates $x$ and $y$ are defined within a bounded region $\Omega$.
- The boundary of the region is $\partial\Omega$. This boundary can be complicated.
- Also, region $\Omega$ can have holes within it defining *interior* boundaries. These interior boundaries are contained in the boundary set $\partial\Omega$.
- Time $t$ is defined over a closed interval, usually $[0, T]$.
- The state of the system at time 0 is $u(x, y; 0) = u_0(x, y)$

# The inhabited portion of the island of Montreal




P. E. Trudeau airport and the rail yards, and the oil refineries and water treatment plant are removed

# Outline

- Function $u$ is the solution of a *partial differential equation*.
- That is, it is defined in terms of an equation relating derivatives with respect $x$, $y$ and $t$.
- These equations can be expressed quite simply in terms of two types of partial derivatives.

We need the “grad” operator $\nabla$, which stands for the operation of calculating the gradient. That is,

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \\ \frac{\partial}{\partial y} \end{bmatrix}$$

so that

$$\nabla u = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \\ \frac{\partial u}{\partial y} \end{bmatrix}.$$

The “Laplacian” operator $\triangle$ can be expressed as $\nabla \cdot \nabla$ and also as $\nabla^2$:

$$\triangle u = \nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

# Outline

The core task of the PDE toolbox is to solve equations of the form:

$$
\begin{aligned}
-\nabla \cdot (\mathbf{C}\nabla u) + au &= f \\
d\frac{du}{dt} - \nabla \cdot (\mathbf{C}\nabla u) + au &= f \\
d\frac{du^2}{d^2t} - \nabla \cdot (\mathbf{C}\nabla u) + au &= f
\end{aligned}
\tag{1}
$$

Scalars $a$, $f$, and $d$; and 2 by 2 matrix $\mathbf{C}$ can be either constants or functions of $t$, $x$, and $y$.

# Interpreting the equations: time dependency

- You can separate the time-dependent behavior from the spatially-dependent behavior.
- The first equation, called “elliptic”, has no time-dependency.
- The second equation, called “parabolic”, has first-order time-dependency, and therefore shows either linear or exponential decay or growth with respect to time.
- The second equation, called “hyperbolic”, has second-order time-dependency, and therefore shows sinusoidal oscillation with respect to time.
- Each of the equations is *forced* by exogenous input represented by function $f$.

# The second-order spatial dependency

- These equations all involve second derivative or curvature variation over space.
- If **C** is constant, the curvature does not vary, and the spatial variation is *isotropic*.
- If **C** is a function, then curvature varies from location to location, and possibly over time as well, and is *anisotropic*.

# The interaction between temporal and spatial variation

- You can think of these equations as ordinary differential equations in time $t$ forced by spatial curvature.
- Let's look at the parabolic equation with constant curvature coefficient $c$ and $f = 0$, taking spatial curvature over to the forcing side:

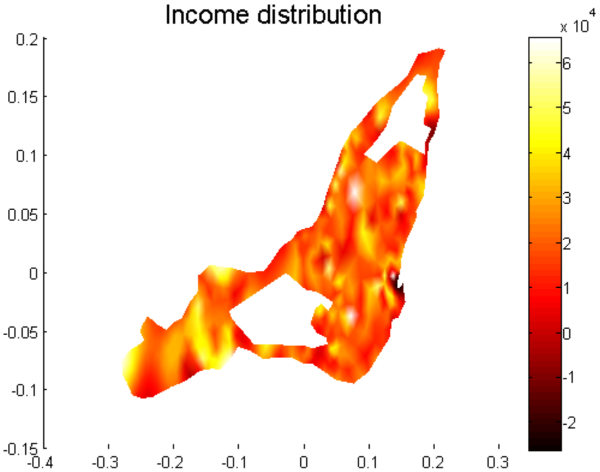
$$d\frac{du}{dt} = -au + c\nabla \cdot \nabla u = -au + c\triangle u$$

- The local rate of change in $u$ over $t$ is proportional to the curvature.

$$d\frac{du}{dt} == au + c\nabla \cdot \nabla u = -au + c\triangle u$$

- If curvature is sharply negative at a location, such as at a peak, $u$ will decay at that location.
- Where the curvature is sharply positive, in a valley or a well, $u$ will increase over time.
- This happens when something *diffuses*, so that high concentrations at a point diffuse outwards, and local low concentrations are increased by inward diffusion.
- This equation is called the *heat* or *diffusion* equation.

# Income in the Island of Montreal



# A socialist scenario

- Quebec separates from Canada, and an aggressively leftist government takes over the legislature.
- Premier Fidèle Castreau and his finance minister, Jacques Léton bring in measures to help the rich share their income with the poor, which they enthusiastically agree to.
- What will happen?

# Elliptic or steady-state equations and hyperbolic or wave equations

- If we replace $du/dt$ in the heat equation by 0, we get the final steady-state that a diffusion process tends to.
- On the other hand, if we replace $du/dt$ by $d^2u/dt^2$ we get the basic linear or exponential decay replaced by oscillation, and this is the *wave* equation.
- In either case, the long-term behavior of $u$ depends on what is happening at the boundaries.
- Let's look at boundary behavior next.

# Outline

# Fixed boundary behavior

- Here are two scenarios.
- Rich people tend to live near the water on the Island of Montreal.
- Perhaps these people will insist that their income be left fixed. Leave it to the people who live in the interior to diffuse their income around.
- Fixing the boundaries is called a *Dirichlet* boundary condition.

$$hu = r \quad \text{on} \quad \partial\Omega$$

  where $h$ and $r$ can be constants or functions.

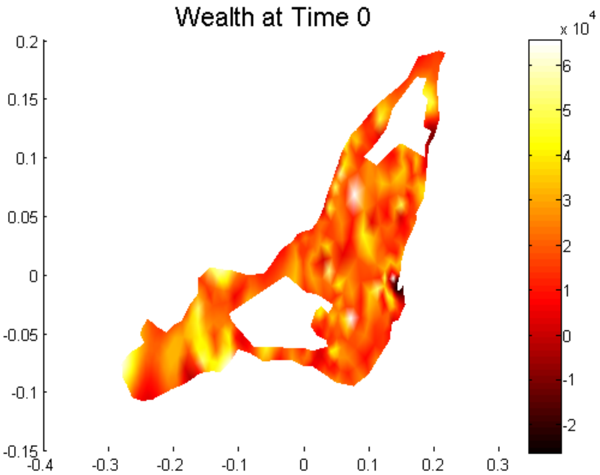
# Fixed flow across the boundary

- On the other hand, perhaps the government will put a wall around the island, so that money can neither enter nor leave. The diffusion across the boundary will be zero.
- This is called a *Neumann* boundary condition.

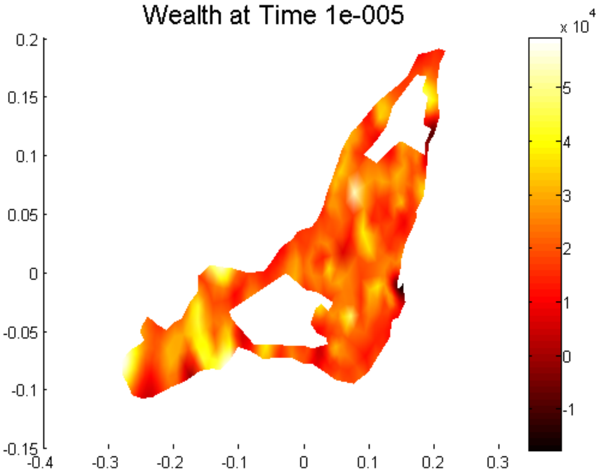
$$\mathbf{n} \cdot (\mathbf{C}\nabla u) + qu = g \quad \text{on} \quad \partial\Omega$$

  where **n** is the normal vector on the boundary, and $g$ is the flow rate across the boundary.
- Mixtures of these conditions are also possible.

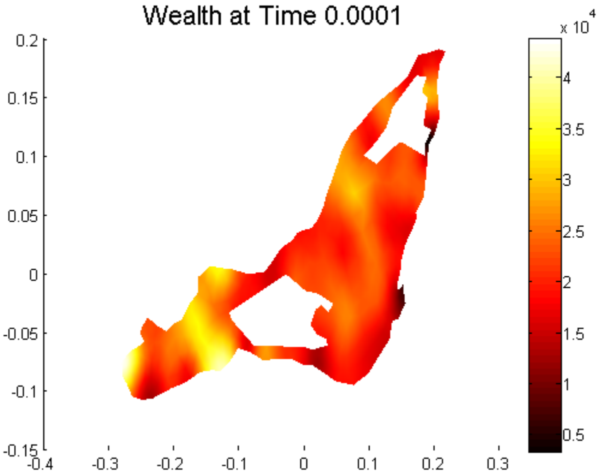
# Outline



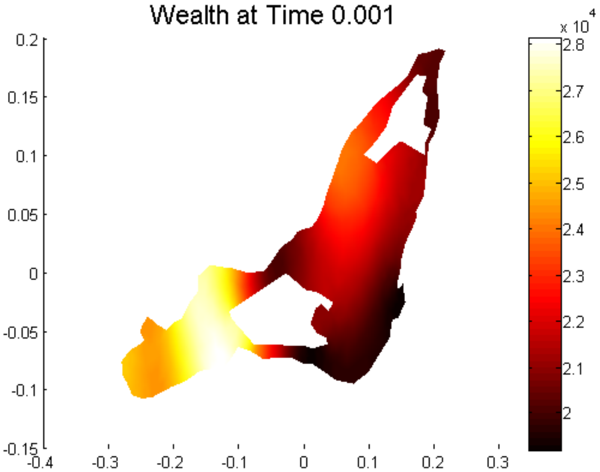

Hampstead has six times the income of Snowdon.




Things are improving; Hampstead has only four times the income of its neighbors.




Near-justice on the mountain, but Beaconsfield has three times the income of Lachine.




Income variation is now much smaller, but income is still piling up behind the airport and rail yards.

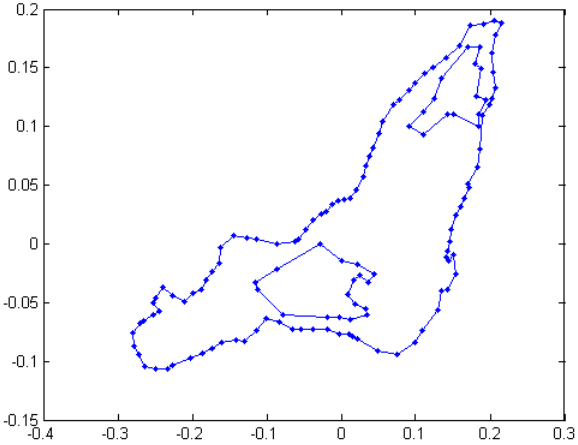
# Spatial smoothing with the heat equation

- The parabolic equation combined with the Neumann boundary condition gives a recipe for spatial smoothing.
- Time $t$ plays the role of the smoothing parameter $\lambda$ in smoothing with a roughness penalty.
- $\triangle u$ plays the role of the roughness penalty. It corresponds to penalty $\int [Dx(t)]^2 \, dt$. You can go from one to the other by integrating by parts.
- The final steady state is a constant, and hence this closely resembles kernel smoothing as well.
- But the Neumann boundary condition plays a key role: it prevents stuff crossing the boundaries. In this case, income cannot cross the airport to get to the other side. Kernel and roughness penalty methods cannot achieve this.

# Outline

# Specifying boundaries

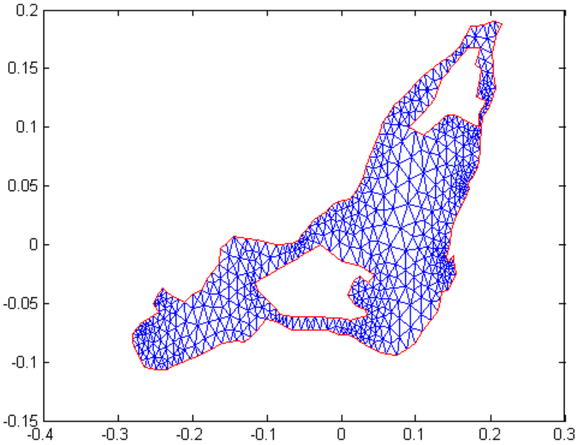
- Boundaries are specified by line and arc segments.
- Boundary information is often available in geographical information system (GIS) databases. I got these boundaries from the Geography library.
- Boundary conditions can be specified separately for each boundary segment.




The outside boundary has 1110 linear segments.

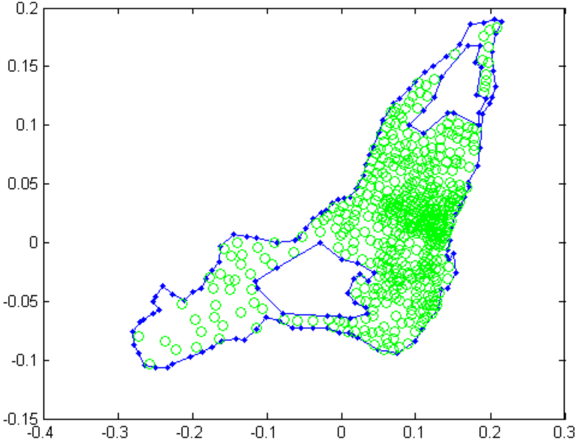
# The interior triangular mesh

- The interior is divided into triangles.
- The sizes of the triangles are adjusted so as to have smaller triangles where things are changing quickly or are spatially complex.
- About 90% of the computational cleverness in the toolbox is associated with getting a good mesh.




There are 1063 triangles covering the area to be smoothed.
They meet at 664 vertices and are defined by 267 edges.

# Census tracts in the Island of Montreal




There were 493 census tracts on the Island.

# Interpolating the information on to the triangular mesh

- The irregularly spaced census income values are interpolated to a rectangular grid.
- Each triangle vertex is assigned the income value of the rectangle containing the vertex.
- This becomes the state of the system at time $t = 0$.

# Finite element basis functions

- A piece-wise linear basis function is associated with each vertex.
- Each interior basis function is nonzero over an irregular hexagon composed of the six triangles that meet at that vertex. Exterior basis functions will have fewer triangular components.
- The basis function is linear everywhere, zero on its exterior boundary, and has value 1 at its central vertex. It looks like a tent.
- The coefficient multiplying each basis function is simply the value of the data at the central vertex.

# Outline

- The differential equation is converted to an equivalent integral equation, and
- this equation is in turn equivalent to a linear equation in the coefficients.
- The linear system is usually huge, but most of the elements of the coefficient matrix are zero. Typically only 5% or so are not. The system is solved using sparse matrix computation methods.

# Outline

- Begin by using the graphical interface for the PDE toolbox.
- To activate it, type `pdetool` into the Matlab command window.
- Work through some of the examples in the manual that comes with the tool box