Very quick introduction to Partial Differential Equations

Pablo Rodriguez-Sanchez

Wageningen University and Research

June 6, 2018

pabrod.github.io/intro-to-pdes-en.html

Disclaimer

These slides are a support for the crash course *Very quick introduction to partial differential equations*, given at Wageningen University in June 2018. As a direct consequence, the slides will perform poorly if used as a self-study material.

Pablo Rodriguez-Sanchez

Outline

What's the difference between EDOs and PDEs?

How are those two equations different?

$$\frac{du}{dx} = -u$$

$$\frac{\partial u}{\partial x} = -u$$

Exercises

Find a function $u(x, y)$ so

$$\frac{\partial u}{\partial x} = x$$

Exercises

Find a function $u(x, y)$ so

$$\frac{\partial u}{\partial x} = x$$

and

$$u(0, y) = \sin y$$

Exercises

Find a function $u(x, y)$ so

$$\frac{\partial u}{\partial x} = x$$

and

$$u(0, y) = \sin y$$

Solution:

$$u(x, y) = \frac{x^2}{2} + \sin y$$

Key ideas

- Derivatives destroy information

Key ideas

- Derivatives destroy information
  - That's why ODE problems need initial conditions to be well posed

Key ideas

- Derivatives destroy information
  - That's why ODE problems need initial conditions to be well posed
- Partial derivatives are weapons of math destruction!

Key ideas

- Derivatives destroy information
  - That's why ODE problems need initial conditions to be well posed
- Partial derivatives are weapons of math destruction!
  - Problem posing with PDEs is hard
  - PDEs are way more powerful

Derivatives in higher dimensions

The *nabla* operator for 2-dimensions is defined as:

$$\vec{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$$

Derivatives in higher dimensions

The *nabla* operator for 2-dimensions is defined as:

$$\vec{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$$

When applied to a *scalar* function, it generates the gradient:

$$\vec{\nabla} u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) = grad\ u$$

Derivatives in higher dimensions

The *nabla* operator for 2-dimensions is defined as:

$$\vec{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$$

When applied to a *scalar* function, it generates the gradient:

$$\vec{\nabla} u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) = grad\ u$$

When applied to a vector function, it generates the divergence:

$$\vec{\nabla} \cdot \vec{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = div\ F$$

Derivatives in higher dimensions

The *nabla* operator for 2-dimensions is defined as:

$$\vec{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$$

When applied to a *scalar* function, it generates the gradient:

$$\vec{\nabla} u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right) = grad\ u$$

When applied to a vector function, it generates the divergence:

$$\vec{\nabla} \cdot \vec{F} = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = div\ F$$

When applied twice, it generates the laplacian:

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

Graphical interpretation of the gradient

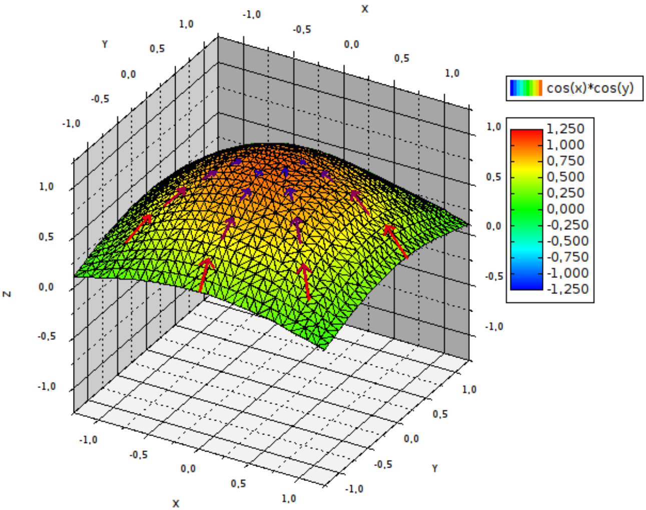

Figure 1: Gradient points to the direction of maximum slope

Graphical interpretation of the divergence

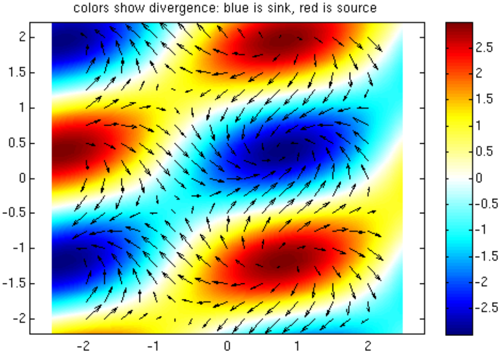

Figure 2: Divergence is maximum in sources and minimum in sinks

Summary of $\vec{\nabla}$

Operator	Shorthand	Explicit
Gradient	$\vec{\nabla} u$	$\left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right)$
Divergence	$\vec{\nabla} \cdot \vec{F}$	$\frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y}$
Laplacian	$\nabla^2 u$	$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$

Table 1: Most common generalizations of derivatives to higher dimensions.

Some classical PDEs

All classical PDEs follow the structure:

$$d\frac{\partial^n u}{\partial t^n} - \vec{\nabla} \cdot (c\vec{\nabla} u) + au = f$$

Some classical PDEs

All classical PDEs follow the structure:

$$d\frac{\partial^n u}{\partial t^n} - \vec{\nabla} \cdot (c\vec{\nabla} u) + au = f$$

n	Type	Some applications
0	Elliptic	Electrostatics, optimization, fluid dynamics
1	Parabolic	Heat and chemical diffusion, quantum mechanics
2	Hyperbolic	Wave motion, electrodynamics

Table 2: Examples of the classical PDEs.

Boundary conditions

In 1-dimension, boundaries just need a beginning and an end

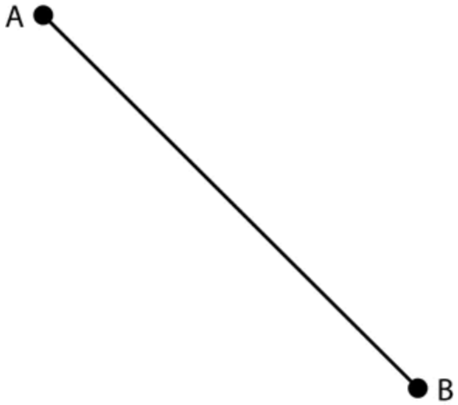

Figure 3: Our region is delimited between A and B.

Boundary conditions

In more dimensions, boundaries have a shape

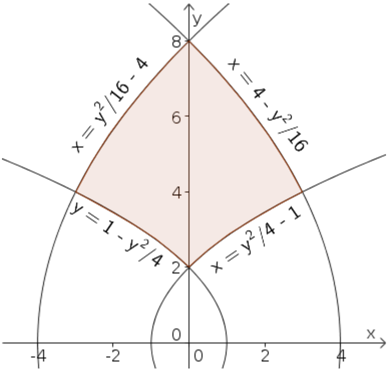

Figure 4: Our region is... well... it's complicated.

Boundary conditions

Classical boundary conditions:

- **Dirichlet**: $u = 0$ in the border

Boundary conditions

Classical boundary conditions:

- **Dirichlet**: $u = 0$ in the border
- **Von Neumann**: directional derivative in the local perpendicular of the border is zero ($\vec{\nabla} u \cdot \vec{n} = 0$)

Boundary conditions

Classical boundary conditions:

- **Dirichlet**: $u = 0$ in the border
- **Von Neumann**: directional derivative in the local perpendicular of the border is zero ($\vec{\nabla} u \cdot \vec{n} = 0$)
- **Periodic**: u is equal at equivalent sides (requires defining what *equivalent sides* means)

Matlab tools

Tool	Dimensions	Form
pdepe	u(x,t)	$c\frac{\partial u}{\partial t} = x^{-m}\frac{\partial}{\partial x}(x^m f) + s$
PDE Toolbox	u(x,y,t)	$d\frac{\partial^n u}{\partial t^n} - \vec{\nabla}\cdot(c\vec{\nabla}u) + au = f$

Table 3: PDE problems and Matlab tool.

pdepe

pdepe is used to solve equations of the following form, where c, f and s are functions of $(x, t, u, \frac{\partial u}{\partial t})$:

$$c\frac{\partial u}{\partial t} = x^{-m}\frac{\partial}{\partial x}(x^m f) + s$$

with boundary conditions:

$$p(x, t, u) + q(x, t) \cdot f = 0$$

Typically, pdepe is used as:

sol = pdepe(m, @pdefun, @icfun, @bcfun, x, t);

Where:

- m is a real number
- x and t are vectors
- c, f, s = pdefun(x, t, u, DuDx)
- u0 = icfun(x)
- pl, ql, pr, qr = bcfun(xl, ul, xr, ur, t)

Cooling rod

In this example we will model step by step a cooling metallic rod. We know that:

- The rod satisfies the heat equation
- The initial temperature of the rod is zero everywhere
- The left border is kept at zero degrees
- The temperature at the right border is forced to oscillate like $u_r(t) = sin(t)$.

Cooling rod

Step 1: find c, f, s so our equation is:

$$\frac{\partial u}{\partial t} = 0.1 \cdot \frac{\partial^2 u}{\partial x^2}$$

Cooling rod

Step 1: find c, f, s so our equation is:

$$\frac{\partial u}{\partial t} = 0.1 \cdot \frac{\partial^2 u}{\partial x^2}$$

Step 2: find pl, ql, pr, qr so our boundary conditions are:

$$u(0, t) = 0$$

$$u(1, t) = sin(t)$$

Cooling rod

Step 1: find c, f, s so our equation is:

$$\frac{\partial u}{\partial t} = 0.1 \cdot \frac{\partial^2 u}{\partial x^2}$$

Step 2: find pl, ql, pr, qr so our boundary conditions are:

$$u(0, t) = 0$$

$$u(1, t) = sin(t)$$

Step 3: set the initial condition $u_0(x) = 0$

Ripple tank

Waves in a water surface (for instance, a swimming pool) follow the wave equation:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u$$

This last exercise will be entirely done with Matlab's PDE toolbox.