On some applied problems using nonlinear elliptic PDEs

C. Finlay

Department of Mathematics and Statistics
McGill University

September 12, 2019

Where do nonlinear elliptic PDEs arise?

- Elliptic PDEs appear in many areas of physics, engineering, economics, computer science ...
- Given differential operator F with certain properties (more on this soon) the general setting is to find a solution u satisfying

$$\begin{cases} F[u(x)] = 0 & (x \in \Omega) \\ u(x) = g & (x \in \partial\Omega) \end{cases} \tag{1}$$

where g is a function defined only on the boundary of Ω.

- Elliptic PDEs behave heuristically like Laplace's equation, which arises when modeling heat flow (diffusion), electrostatics, fluid dynamics...

$$\begin{cases} -\nabla \cdot \nabla u(x) = 0 & (x \in \Omega) \\ u(x) = g & (x \in \partial\Omega) \end{cases}$$

Where do nonlinear elliptic PDEs arise?

- A nonlinear example: the Hamilton-Jacobi-Bellman operator

$$\begin{cases} \sup_\alpha \{\mathcal{L}^\alpha u(x)\} = 0 & (x \in \Omega) \\ u(x) = g & (x \in \partial\Omega) \end{cases}$$

  Comes from optimal control of stochastic processes (finance, electrical engineering, management, Markov processes)
- And many other areas: optimal transport, image processing, differential games, semi-supervised learning. . .

Recognizing nonlinear elliptic PDEs in the wild

Nonlinear elliptic PDEs satisfy a *weak comparison principle*: given two functions u and v, an operator F is elliptic if

$$\begin{aligned} F[u] \leq F[v] \quad (x \in \Omega) \\ \implies u \geq v \quad (x \in \Omega) \end{aligned}$$

- Unfortunately classical solutions (those that are twice differentiable) don't necessarily exist for nonlinear elliptic PDEs
- Traditional weak solution techniques fail here because nonlinear equations don't have a divergence structure to exploit. We can't pass derivatives onto a test function using integration by parts.

Viscosity solutions

Instead, use the notion of a *viscosity solution*.

- Requirements of differentiability are passed onto smooth test functions ϕ that graze a candidate solution u from above (or below).
- If the test function ϕ grazes from above at x, and $F[\phi(x)] \leq 0$, then u is a *viscosity sub-solution*.
- Similarly we can define super-solutions
- A viscosity solution is both a sub- and super-solution.

Viscosity solutions are the theoretical framework of choice for proving existence, uniqueness and regularity results for nonlinear elliptic PDEs.

Application: Homogenization

In certain environments, the operator $F^{\varepsilon}[u^{\epsilon}]$ and its solution u^{ϵ} is highly oscillatory, depending on a microscopic scale parameter ε.

- We often only care about the macroscopic behaviour (eg composite materials).
- want a macroscopic operator $F[u]$ which is a limiting PDE as $\varepsilon \to 0$, with solutions converging uniformly $u^{\varepsilon} \to u$
- Evans [Eva89,Eva92] showed the homogenized operator can be found using perturbed viscosity test functions by solving a "cell problem"
- Chapters 2 & 3 of the thesis deal with approximate methods for analytic solutions of the cell problem

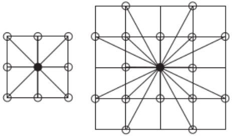
Numerical solutions and monotone schemes

In practice, we can only compute *approximate* viscosity solutions, numerically.

- Our numerical schemes must be *provably* convergent. As we increase our computational effort, we need to know our computed solution approaches the true analytic solution.
- For viscosity solutions, this is done using the Barles and Sougandidis framework [BS91]. A numerical scheme for an elliptic PDE is convergent if
  1. it respects the underlying PDE's comparison principle (it must be monotone increasing)
  2. it is stable (small perturbations don't yield vastly different results)
  3. is consistent (the error of the numerical operator decreases with more computational effort)

However *a priori* it is not at all obvious how to build a numerical scheme satisfying these three components.

Monotone elliptic schemes from finite differences

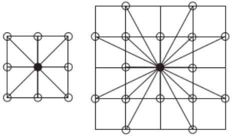


Wide stencil finite difference schemes (source: [Ob08])

Fortunately many nonlinear elliptic PDEs may be interpreted geometrically as being composed of directional derivatives.

- This leads to building so-called elliptic schemes [Ob06,Ob08] in which directional derivatives are approximated with finite differences
- Moreover, elliptic schemes satisfy the Barles and Sougandidis framework, so convergence is guaranteed

Example: the maximum eigenvalue of the Hessian



Wide stencil finite difference schemes (source: [Ob08])

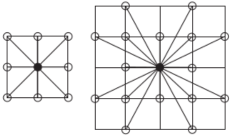
Suppose we want to solve $\lambda_1[D^2u(x)] = 0$, where D^2u is the Hessian matrix of second derivatives, and $\lambda_1[\cdot]$ is the maximum eigenvalue.

Recall that the maximum eigenvalue of a matrix is given by $\lambda_1[D^2u] = \max_{\|v\|=1}\langle v, (D^2u)v\rangle$.

This is just a maximum of directional derivatives: $\max_v \frac{\partial^2 u}{\partial v^2}$

- approximate $\frac{\partial^2 u}{\partial v^2} \approx \frac{1}{h^2}\left[u(x+hv) - 2u(x) + u(x-hv)\right]$
- approximate the maximum by only using directions v available on the grid

Balancing angular and spatial resolution

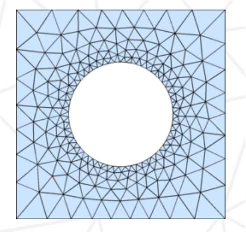


Wide stencil finite difference schemes (source: [Ob08])

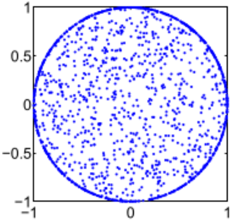
- On the one hand, we want many search directions v to better approximate $\max_v$. More search directions leads to better *angular resolution* $d\theta$.
  - Leads to wider and wider stencils
- On the other hand, the stencil can't be too wide: wide stencils degrade the finite difference error, which depends on *spatial resolution* h

Rhetorical question: Wouldn't it be nice to have off-grid search directions?

Irregular grids and point clouds



An irregular grid (source: distmesh)



A point cloud (source: [Fro18])

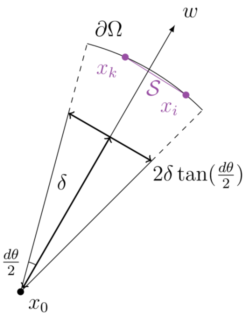
Moreover, what happens when our data doesn't lie on a rectangular grid? In many real-world applications, data has either (i) a graph structure, or (ii) *no structure at all*

- No search directions lie on an irregular grid
- The symmetric finite difference scheme

$$\frac{1}{h^2}\left[u(x+hv)-2u(x)+u(x-hv)\right]$$

isn't available

Our solution: finite differences with linear interpolation

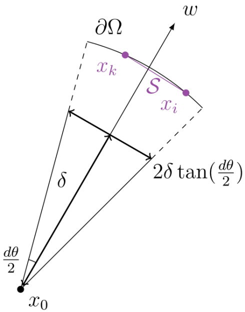


Finite differences with interpolation

We overcome these problems with linear interpolation between available points. For example suppose we want the directional derivative $\frac{\partial u}{\partial w}$ at the point x_0

- we first approximate $\frac{\partial u}{\partial w} \approx \frac{1}{h}\left[u(x_0 + hw) - u(x_0)\right]$
- since $x_0 + hw$ is *not* an available point, we interpolate between nearest neighbours x_k and x_i (in purple on figure) $u(x_0 + hw) \approx L[u(x_k), u(x_i)]$
- Leads to the approximation $\frac{\partial u}{\partial w} \approx \frac{1}{h}\left(L[u(x_k), u(x_i)] - u(x_0)\right)$

Finite differences with linear interpolation are convergent



Finite differences with interpolation

We can show that

- These schemes are consistent: the linear interpolation error can be controlled
- They are monotone and stable: linear interpolation respects monotonicity and stability

Hence Barles and Sougandidis' framework for convergence can be used. Moreover can show that

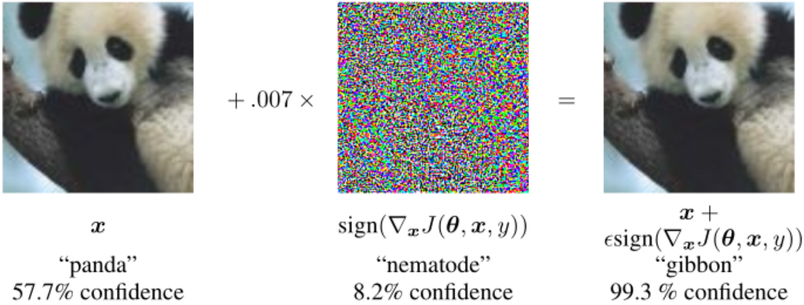
- The schemes exist on both interior points and near the boundary, in any dimension

Comparison of discretization methods

Scheme	Order	Optimal $d\theta$	Formal accuracy	Comments
Nearest grid direction [Ob08]	$\mathcal{O}(R^2 + d\theta)$	$\mathcal{O}(h^{\frac{2}{3}})$	$\mathcal{O}(h^{\frac{2}{3}})$	Regular grids. Difficult implementation near boundaries.
Two-scale convergence [NNZ19]	$\mathcal{O}(R^2+d\theta^2)$	$\mathcal{O}(h^{\frac{1}{2}})$	$\mathcal{O}(h)$	n-d, for triangulations. Consistent away from boundary.
Froese [Fro18]	$\mathcal{O}(R + d\theta)$	$\mathcal{O}(h^{\frac{1}{2}})$	$\mathcal{O}(h^{\frac{1}{2}})$	2d, mesh free. No difficulty at boundary.
Linear interpolant, symmetric	$\mathcal{O}(R^2+d\theta^2)$	$\mathcal{O}(h^{\frac{1}{2}})$	$\mathcal{O}(h)$	n-d, regular grids. No difficulty at boundary.
Linear interpolant, non symmetric	$\mathcal{O}(R + d\theta^2)$	$\mathcal{O}(h^{\frac{1}{3}})$	$\mathcal{O}(h^{\frac{2}{3}})$	n-d, mesh free. No difficulty at boundary.

Stability in Neural Networks

Neural networks used in image classification are vulnerable to *adversarial attacks*. In other words, they are unstable: small changes in input yield to wildly different predictions.



An *adversarial example* in image classification (source: [GSS14])

Gradient regularization

In supervised learning, the objective is to find a function $u(x;\theta)$ parameterized by θ which minimizes a loss. In regression the squared L^2 error is minimized:

$$\min_\theta \int (u(x;\theta) - f(x))^2 \,\mathrm{d}\rho$$

If we want the learned function u to be robust to perturbations, heuristically it makes sense to penalize u for large gradients

$$\min_\theta \int (u(x;\theta) - f(x))^2 + \lambda \|\nabla_x u(x;\theta)\|^2 \,\mathrm{d}\rho \tag{2}$$

This is called Tikhonov regularization and is used heavily in inverse problems.

- Euler-Lagrange for (2) is the elliptic PDE

$$u - \frac{1}{\rho}\nabla \cdot (\rho \nabla u) = f$$

Bounds on perturbation size justify gradient regularization

We can show that neural networks with small gradients are provably robust to adversarial perturbations in image classification problems.

- If the neural network is continuous but not differentiable (usually the case) then we can bound the minimum adversarial perturbation size by the maximum gradient of the network (its Lipschitz constant)
- If the neural network is differentiable, we show a tighter bound on minimum perturbation size by the gradient at x and a curvature bound

In other words, gradient regularization will promote robustness.

How to implement the gradient penalty?

It is not feasible to solve the Euler-Lagrange equations in high dimensions, so instead people minimize the loss directly.
With our gradient penalty, during the optimization process we will need to calculate

$$\nabla_\theta \|\nabla_x u(x;\theta)\|^2$$

- naive approach: use automatic differentiation twice, once in x, then again in θ.
  - Unfortunately this is slow and does not scale to real-world networks.

Finite differences, again

Instead we use finite differences, which do scale to large networks like those used on ImageNet-1k.

- First compute $d = \frac{\nabla_x u}{\|\nabla_x u\|}$ using automatic differentiation, and detach it from the "computational graph"
- A simple Taylor series expansion gives the approximation

$$\|\nabla_x u\| \approx \frac{1}{h}\left[u(x + hd) - u(x)\right]$$

- We then estimate

$$\begin{aligned}&\nabla_\theta \|\nabla_x u(x;\theta)\|^2 \\ &\approx \frac{2}{h}\left[u(x+hd) - u(x)\right]\left(\nabla_\theta u(x+hd;\theta) - \nabla_\theta u(x;\theta)\right)\end{aligned}$$

To our knowledge, this is the first scaleable technique for adversarial robustness on ImageNet-1k.

Take home message

- Elliptic PDEs arise naturally when modeling many systems
- In low dimension they can be solved accurately, even on unstructured point clouds
- Though solving an elliptic PDE may not tractable in high dimensions, techniques from the numerical analysis and PDE literature can guide and motivate high dimensional algorithms

References I

- Guy Barles and Panagiotis E. Souganidis, *Convergence of approximation schemes for fully nonlinear second order equations*, Asymptotic Anal. **4** (1991), no. 3, 271–283.
- Lawrence C Evans, *The perturbed test function method for viscosity solutions of nonlinear PDE*, Proceedings of the Royal Society of Edinburgh: Section A Mathematics **111** (1989), no. 3-4, 359–375.
- Lawrence C Evans, *Periodic homogenisation of certain fully nonlinear partial differential equations*, Proceedings of the Royal Society of Edinburgh: Section A Mathematics **120** (1992), no. 3-4, 245–265.
- Brittany D. Froese, *Meshfree finite difference approximations for functions of the eigenvalues of the Hessian*, Numerische Mathematik **138** (2018), no. 1, 75–99.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, *Explaining and harnessing adversarial examples*, CoRR **abs/1412.6572** (2014).

References II

R Nochetto, Dimitrios Ntogkas, and Wujun Zhang, *Two-scale method for the monge-ampère equation: Convergence to the viscosity solution*, Mathematics of Computation **88** (2019), no. 316, 637–664.

Adam M. Oberman, *Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton-Jacobi equations and free boundary problems*, SIAM J. Numer. Anal. **44** (2006), no. 2, 879–895 (electronic).

Adam M. Oberman, *Wide stencil finite difference schemes for the elliptic Monge-Ampère equation and functions of the eigenvalues of the Hessian*, Discrete Contin. Dyn. Syst. Ser. B **10** (2008), no. 1, 221–238.

Per-Olof Persson and Gilbert Strang, *A simple mesh generator in MATLAB*, SIAM review **46** (2004), no. 2, 329–345.