Solutions of Semilinear Elliptic PDEs on Manifolds

Jeff Springer, Northern Arizona University

What is a PDE?

Definition

A Partial Differential Equation (PDE) is a relation involving an unknown function of several independent variables and their partial derivatives with respect to those variables.

PDEs on Surfaces

- For today's talk we will be ultimately concerned with solving PDEs (in some fashion) on surfaces.
- Many applications of solving PDEs on surfaces: image processing, flow and transport in earth's oceans, etc.
- We are primarily interested in solving surface eigenvalue problems which has many specific applications.
- Eigenvalue problems on surfaces lead to applications in design optimization, shape recognition, and quantum billiards.

Introduction

- We are interested in solving the problem:

$$\Delta_S u + f(u) = 0$$

- In the sequel we choose the particular nonlinearity $f(u) = su + u^3$.
- We solve this PDE on a manifold $W \subseteq \mathbb{R}^n$, where Δ_S above denotes the Laplace-Beltrami operator.
- Much of the well-known theory for the PDE $\Delta u + f(u) = 0$ extends nicely to solving this problem on manifolds.

Introduction II

- Taking a variational approach, we introduce the action functional: $J : H^{(1,2)}(W) \to \mathbb{R}$.
- The action functional J satisfies the properties:

$$\text{J(u)} = \int_W \frac{1}{2}|\nabla u|^2 - F(u)ds$$

$$\text{J'(u)(v)} = \langle \nabla J(u), v \rangle = \int_W \{\nabla u \cdot \nabla v - f(u)v\}ds$$

$$\text{J''(u)(v,w)} = \int_W \{\nabla v \cdot \nabla w - f'(u)vw\}ds$$

A theorem

- $\nabla J(u) = 0$ if and only if ,u is a classical solution of $\Delta u + f(u) = 0$. Hence, the solutions to this PDE are critical points of the action functional
- First, we solve the eigenvalue problem: $\Delta\psi_i = \lambda_i\psi_i$ on W. This is done with known closed-form eigenfunctions or CPM.
- Next we let $u = \sum a_i\psi_i$.

The ψ matrix

$$\Psi = \begin{bmatrix} \psi_1(\frac{1}{n}) & \psi_2(\frac{1}{n}) & \cdots & \psi_M(\frac{1}{n}) \\ \psi_1(\frac{2}{n}) & \psi_2(\frac{2}{n}) & \cdots & \psi_M(\frac{2}{n}) \\ \vdots & \vdots & \vdots & \vdots \\ \psi_1(\frac{n-1}{n}) & \psi_2(\frac{n-1}{n}) & \cdots & \psi_M(\frac{n-1}{n}) \end{bmatrix}$$

GNGA

The Gradient Newton Galerkin Algorithm (GNGA) is an M dimensional Newtons method applied to an approximated gradient function $g : \mathbb{R}^M \to \mathbb{R}^M$ defined by

$$g = \begin{bmatrix} J'(u)(\psi_1) \\ \vdots \\ J'(u)(\psi_M) \end{bmatrix} = \begin{bmatrix} \int_W (\nabla u \cdot \nabla \psi_1 - su\psi_1 - u^3\psi_1)dx \\ \vdots \\ \int_W (\nabla u \cdot \nabla \psi_M - su\psi_M - u^3\psi_M)dx \end{bmatrix}$$

Derivation of Gradient

We define the gradient, as on the previous slide, as $g_i = J'(u)(\psi_i)$. Thus $g_i \in \mathbb{R}^M$. The gradient can be simplified as:

$$\begin{aligned} J'(u)(\psi_i)) &= \int_W (\nabla u \cdot \nabla \psi_i - su\psi_i - u^3\psi_i)dx \\ &= \int_W (\nabla \Sigma a_k \psi_k \cdot \nabla \psi_i - su\psi_i - u^3\psi_i)dx \\ &= (\lambda_i - s)a_i - \int_W u^3\psi_i)dx \end{aligned}$$

Note that $(\Psi' * \frac{u^3}{n} \approx \int_W u^3\psi_i)dx$

The Hessian matrix

We use a Jacobian of g as the derivative in the algorithm. We compute this approximated Hessian h as:

$$\mathrm{h}_{ij} = ((\lambda_i - s)\delta_{ij} - 3\int_W u^2 \psi_i \psi_j dx)$$

where $i, j = \{1, 2, \ldots, M\}$

Derivation of Hessian

To obtain the simplified version of the Hessian on the previous slide we use the definition of the Hessian:

$$\begin{aligned} h_{ij} &= J''(u)(\psi_i, \psi_j) \\ &= \int_W (\nabla \psi_i \cdot \nabla \psi_j - s\psi_i\psi_j - 3u^2\psi_i\psi_j)dx \\ &= (\lambda_i - s)\delta_{ij} - \int_W (3u^2\psi_i\psi_j)dx \end{aligned}$$

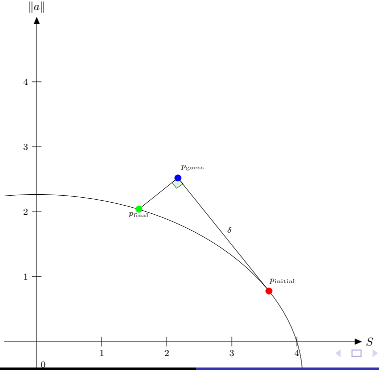
GNGA Algorithm

1. Choose your initial coefficients $a = a^0 = \{a_k\}_{k=1}^M$, set $u = u^0 = \sum a_k \psi_k$, and set $n = 0$
2. Loop over:
    1. Calculate $g = g^{n+1} = (J'(u)(\psi_k))_{k=1}^M \in \mathbf{R}^\mathbf{M}$ (gradient).
    2. Calculate $A = A^{n+1} = (J''(u)(\psi_j, \psi_k))_{j,k=1}^M$ (Hessian).
    3. Compute $\chi = \chi^{n+1} = A^{-1} g$ by solving the system.
    4. Set $a = a^{n+1} = a^n - \delta\chi$ and update $u = u^{n+1} = \sum a_k \psi_k$.
    5. Increment counter n
    6. Calculate $\mathrm{sig}(A(a))$,if desired.
    7. STOP when $\sqrt{g \cdot g} = \|P_G \nabla J(u)\| < \mathrm{TOL}$.

tGNGA

- Create tangent vector $T = \frac{(P_{\text{cur}} - P_{\text{old}})}{\|P_{\text{cur}} - P_{\text{old}}\|}$. Define initial guess $P_{\text{guess}} = P_{\text{cur}} + \delta T$.
- We then define the constraint $\kappa = (P - P_{\text{guess}}) \cdot T$.
- Forcing $\kappa = 0$ ensures that the vector made by our solution P and P_{guess} is perpendicular to T.
- Note that $p = (a_i, s) \in \mathbb{R}^{\mathbb{M}+\mathbb{1}}$

Illustration of tGNGA

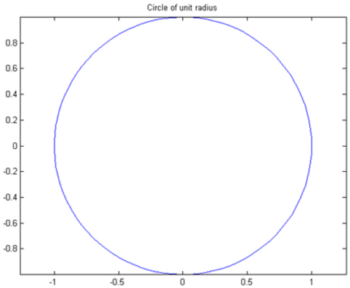


cGNGA

- Given a solution p^* where the k^{th} eigenvalue is zero, let $\hat{e}_k$ be the corresponding eigenfunction of the Hessian.
- We define the constraint $\kappa = ||P_{\hat{e}_k} u||^2 - \delta^2$.
- Forcing this to be zero ensures that the projection of our solution u onto the k^{th} eigenvector is some positive amount δ.
- If this is symmetry breaking then u not on the mother branch.

The ring: $x^2 + y^2 = 1$

$\Delta_S u + f(u) = 0$ on the unit ring:



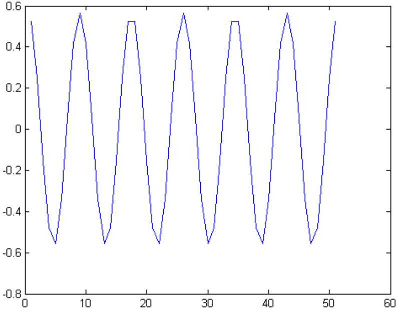
Psi matrix for the ring

The Psi matrix for the ring is given by:

$$\Psi = \begin{bmatrix} \vdots & \vdots & \vdots \\ \frac{1}{\sqrt{2\pi}} & \frac{\cos(m\theta_i)}{\sqrt{\pi}} & \frac{\sin(m\theta_i)}{\sqrt{\pi}} \\ \vdots & \vdots & \vdots \end{bmatrix}$$

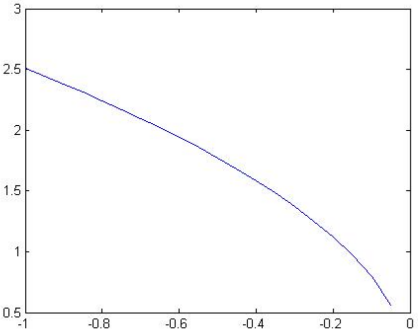
Eigenfunctions on the ring

Here is a picture of the seventh eigenfunction on the ring:



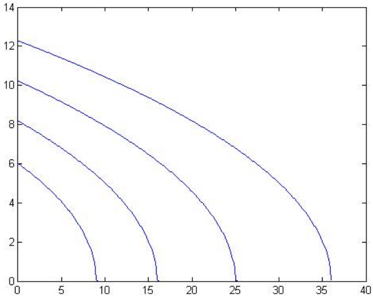
Constant branch on the ring

Here is a picture of the constant bifurcation branch for the ring:



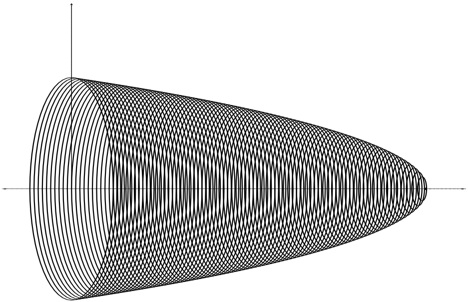
First four nontrivial branches on the ring

Here is a picture of the first four non-trivial bifurcation branches for the ring:



Example of Bifurcation Cone on the ring

Since any rotation of a solution is also a solution to the PDE we obtain a bifurcation cone for each branch:



The Sphere

- Our next experiment is to solve the PDE on the sphere.
- Spherical harmonics are the well-known eigenfunctions of Δ_S on the sphere.
- The spherical harmonics look like: $Y_l^m(\psi, \theta) = P_l^m(\cos(\phi)(a cos(m\theta) + b\sin(m\theta))$, where P_l^m, $|m| \leq l$ is the associated Legendre function.

Spherical Harmonics

Below is a picture illustrating the real part of several lower order/degree spherical harmonics:

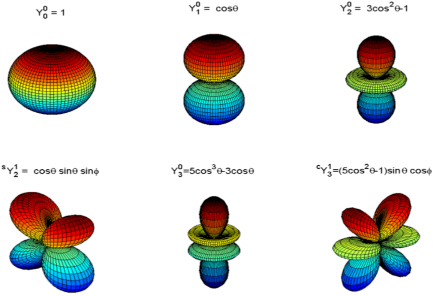


Figure: Low order/degree real spherical harmonics

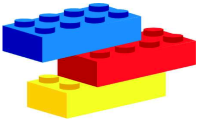
Next steps for Sphere

- Currently writing code to populate ψ matrix for sphere.
- Next step to implement tGNGA/cGNGA code on the sphere
- Symmetry analysis of the solutions on the sphere

No closed form eigenfunctions?

- What to do if we do not know closed form for eigenfunctions?
- Use the Closest Point Method!
- The Closest Point Method (CPM) is a recent embedding method for solving time-dependent PDE's on surfaces, which can also easily be utilized to solve eigenvalue problems on surfaces.
- With CPM we can solve our PDE of interest on general manifolds, even those without a well defined inside/outside!

Building blocks of CPM



Closest Point Method uses three simple and fundamental "building blocks" of numerical analysis:

1. Interpolation
2. Finite differences
3. Time stepping

These are combined in a straightforward way to solve surface eigenvalue problems.

Details of the CPM

- Formally, if S is a smooth surface in $\mathbb{R}^d$then the closest point extension of $v : S \to \mathbb{R}$ is a function $u : \Omega \to \mathbb{R}$, defined in a neighborhood $\Omega \subset \mathbb{R}^d$ of S,as $u(x) = v(cp(x))$. We say that u is a closest point extensionof v.
- We use CPM to solve the problem: Given a surface W determine a surface eigenfunction $u : S \to \mathbb{R}$ and eigenvalue λ such that:

$$-\Delta_S(u(x)) = \lambda u(x)$$

An embedded eigenvalue problem

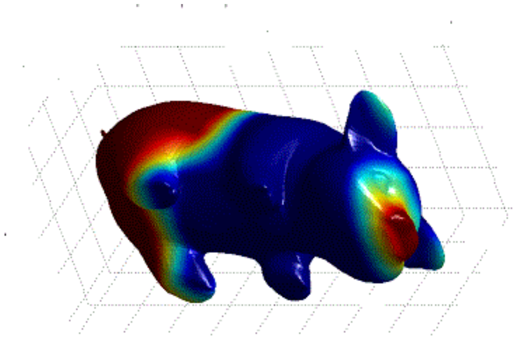
- We first would like to solve the problem: Determine the eigenfunctions $v : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ and eigenvalues λ satisfying:

$$-\Delta(v(\mathrm{cp}(x))) = \lambda v(x)$$

- This eigenvalue problem is ill-posed. Why?
- The set of null-eigenfunctions of the embedded eigenvalue problem is much larger than the set of null-eigenfunctions for the Laplace-Beltrami eigenvalue problem.

An eigenfunction on the Pig using CPM

- Here is a picture of an eigenfunction on the surface of the pig, obtained using the CPM:



Future Work

- Finalize computations and bifurcation diagram for sphere
- Study more interesting manifolds, like Anne's Pig above, using CPM.
- Compute bifurcation diagrams for surfaces with mixed co-dimension.
- Continue to generalize the well known variational theory to solving on manifolds.

Kaust Beacon

Here is one interesting manifold I am interested in solving our PDE on in the near future:

Acknowledgments

I would like to thank the following people:

- Dr. John Neuberger
- Dr. Jim Swift
- Tyler Diggans
- Ian Douglas

Questions?

Thank you for listening!