Numerical solutions of PDEs on Manifolds

Parousia Rockstroh

Harvey Mudd College

23 April 2008

PDEs in Math 164

- In Math 164 we've worked with implicit/explicit methods for solving PDEs in $\mathbb{R}^n$.

PDEs in Math 164

- In Math 164 we've worked with implicit/explicit methods for solving PDEs in $\mathbb{R}^n$.
- We've also considered PDEs on other domains which are subsets of $\mathbb{R}^2$ (e.g., the wave equation on a circle).

PDEs in Math 164

- In Math 164 we've worked with implicit/explicit methods for solving PDEs in $\mathbb{R}^n$.
- We've also considered PDEs on other domains which are subsets of $\mathbb{R}^2$ (e.g., the wave equation on a circle).
- Is it possible to solve PDEs on more complex surfaces?

PDEs on surfaces: Why we care

PDEs on surfaces: Why we care

- Problems often arise in which PDEs must be solved on complex surfaces.

PDEs on surfaces: Why we care

- Problems often arise in which PDEs must be solved on complex surfaces.
  - Biological systems (solution of predator-prey on surfaces)

PDEs on surfaces: Why we care

- Problems often arise in which PDEs must be solved on complex surfaces.
    - Biological systems (solution of predator-prey on surfaces)
    - Image Processing (generation of textures)

PDEs on surfaces: Why we care

- Problems often arise in which PDEs must be solved on complex surfaces.
  - Biological systems (solution of predator-prey on surfaces)
  - Image Processing (generation of textures)
  - Medical imaging (brain scans)

PDEs on surfaces: Why we care

- Problems often arise in which PDEs must be solved on complex surfaces.
  - Biological systems (solution of predator-prey on surfaces)
  - Image Processing (generation of textures)
  - Medical imaging (brain scans)
  - Fluid dynamics (flows and solidification on surfaces)

PDEs on surfaces: Why we care

- Problems often arise in which PDEs must be solved on complex surfaces.
  - Biological systems (solution of predator-prey on surfaces)
  - Image Processing (generation of textures)
  - Medical imaging (brain scans)
  - Fluid dynamics (flows and solidification on surfaces)
  - David Coat's problem (terrain decay)

PDEs on surfaces: Why we care

- Problems often arise in which PDEs must be solved on complex surfaces.
    - Biological systems (solution of predator-prey on surfaces)
    - Image Processing (generation of textures)
    - Medical imaging (brain scans)
    - Fluid dynamics (flows and solidification on surfaces)
    - David Coat's problem (terrain decay)
- It is an interdisciplinary problem that involves a great deal of computer science, graphics, and math.

An initial approach: Triangulation

An initial approach: Triangulation

The method of triangulation

An initial approach: Triangulation

The method of triangulation

- Approximate the given surface by triangulating.

An initial approach: Triangulation

The method of triangulation

- Approximate the given surface by triangulating.
- Solve the given PDE on each triangle

An initial approach: Triangulation

The method of triangulation

- Approximate the given surface by triangulating.
- Solve the given PDE on each triangle
- Piece the solutions together

An initial approach: Triangulation

The method of triangulation

- Approximate the given surface by triangulating.
- Solve the given PDE on each triangle
- Piece the solutions together
- If this is not good enough, then use more triangles!

An initial approach: Triangulation

The method of triangulation

- Approximate the given surface by triangulating.
- Solve the given PDE on each triangle
- Piece the solutions together
- If this is not good enough, then use more triangles!

Issues with triangulation

An initial approach: Triangulation

The method of triangulation

- Approximate the given surface by triangulating.
- Solve the given PDE on each triangle
- Piece the solutions together
- If this is not good enough, then use more triangles!

Issues with triangulation

- We have to use a lot of triangles! ($\sim 10^6$)

An initial approach: Triangulation

The method of triangulation

- Approximate the given surface by triangulating.
- Solve the given PDE on each triangle
- Piece the solutions together
- If this is not good enough, then use more triangles!

Issues with triangulation

- We have to use a lot of triangles! ($\sim 10^6$)
- It's hard to splice solutions together.

An initial approach: Triangulation

The method of triangulation

- Approximate the given surface by triangulating.
- Solve the given PDE on each triangle
- Piece the solutions together
- If this is not good enough, then use more triangles!

Issues with triangulation

- We have to use a lot of triangles! ($\sim 10^6$)
- It's hard to splice solutions together.
- Convergence of numerical schemes on triangulated grids are not understood as well as with Cartesian grids.

An initial approach: Parametrization

Solving PDEs via parameterization

An initial approach: Parametrization

Solving PDEs via parameterization

- Parameterize the given surface

An initial approach: Parametrization

Solving PDEs via parameterization

- Parameterize the given surface
- Conformally map it to a cartesian grid

An initial approach: Parametrization

Solving PDEs via parameterization

- Parameterize the given surface
- Conformally map it to a cartesian grid
- Solve the problem in Cartesian coordinates!

An initial approach: Parametrization

Solving PDEs via parameterization

- Parameterize the given surface
- Conformally map it to a cartesian grid
- Solve the problem in Cartesian coordinates!

Issues with this approach

An initial approach: Parametrization

Solving PDEs via parameterization

- Parameterize the given surface
- Conformally map it to a cartesian grid
- Solve the problem in Cartesian coordinates!

Issues with this approach

- Some surfaces are hard to parametrize!

An initial approach: Parametrization

Solving PDEs via parameterization

- Parameterize the given surface
- Conformally map it to a cartesian grid
- Solve the problem in Cartesian coordinates!

Issues with this approach

- Some surfaces are hard to parametrize!
- Finding a good conformal map can be quite challenging!

An initial approach: Parametrization

Solving PDEs via parameterization

- Parameterize the given surface
- Conformally map it to a cartesian grid
- Solve the problem in Cartesian coordinates!

Issues with this approach

- Some surfaces are hard to parametrize!
- Finding a good conformal map can be quite challenging!

Is there another approach to transforming our problem to cartesian coordinates?

- The Closest Point Method gives us an easy and efficient way of projecting the problem into Cartesian coordinates, thereby allowing us to solve it with classical numerical schemes.

- The Closest Point Method gives us an easy and efficient way of projecting the problem into Cartesian coordinates, thereby allowing us to solve it with classical numerical schemes.
- We don't need triangulations!

- The Closest Point Method gives us an easy and efficient way of projecting the problem into Cartesian coordinates, thereby allowing us to solve it with classical numerical schemes.
- We don't need triangulations!
- We don't need conformal mappings!

- The Closest Point Method gives us an easy and efficient way of projecting the problem into Cartesian coordinates, thereby allowing us to solve it with classical numerical schemes.
- We don't need triangulations!
- We don't need conformal mappings!
- We will assume that we are working with a manifold of codimension-one that is embedded in $\mathbb{R}^n$.

- The Closest Point Method gives us an easy and efficient way of projecting the problem into Cartesian coordinates, thereby allowing us to solve it with classical numerical schemes.
- We don't need triangulations!
- We don't need conformal mappings!
- We will assume that we are working with a manifold of codimension-one that is embedded in $\mathbb{R}^n$.
- The Closest Point Method utilizes the geometry of the surface.

Signed Distance Function

Signed Distance Function

Definition (Signed Distance Function)

Let S be a closed embedded codimension-one manifold in $\mathbb{R}^n$. A function $\phi : R^n \rightarrow R$ is a signed distance function if $\phi < 0$ inside S, and $\phi > 0$ outside S.

Signed Distance Function

Definition (Signed Distance Function)

Let S be a closed embedded codimension-one manifold in $\mathbb{R}^n$. A function $\phi : R^n \rightarrow R$ is a signed distance function if $\phi < 0$ inside S, and $\phi > 0$ outside S.

Example

- The signed distance function for the circle is given by:

$$\phi(x, y) = \sqrt{x^2 + y^2} - 1.$$

Signed Distance Function

Definition (Signed Distance Function)

Let S be a closed embedded codimension-one manifold in $\mathbb{R}^n$. A function $\phi : R^n \to R$ is a signed distance function if $\phi < 0$ inside S, and $\phi > 0$ outside S.

Example

- The signed distance function for the circle is given by:

$$\phi(x, y) = \sqrt{x^2 + y^2} - 1.$$

- The signed distance function for the torus is given by:

$$\phi(x, y, z) = \sqrt{z^2 + \sqrt{x^2 + y^2} - R} - r$$

Computational Domain

Computational Domain

- We need to defin a domain on which to do our computations.

Computational Domain

- We need to defin a domain on which to do our computations.
- Let c be a constant such that:

$$|\phi(x)| \leq c < \frac{1}{\kappa \text{max}}$$

Computational Domain

- We need to defin a domain on which to do our computations.
- Let c be a constant such that:

$$|\phi(x)| \leq c < \frac{1}{\kappa_{\max}}$$

- We will define our computational domain as the following band:

$$\Omega_c = \{x : |\phi(x)| \leq c\}.$$

Computational Domain

- We need to defin a domain on which to do our computations.
- Let c be a constant such that:

$$|\phi(x)| \leq c < \frac{1}{\kappa_{\max}}$$

- We will define our computational domain as the following band:

$$\Omega_c = \{x : |\phi(x)| \leq c\}.$$

- The set $\Omega_c = \{x : |\phi(x)| = 0\}$ is called the zero level set of $\phi(x)$, and defines the manifold.

The Projection Matrix

The Projection Matrix

The Projection Matrix

The Projection Matrix

The Projection Matrix

- We need some way of getting from the set Ω_c to $\mathbb{R}^n$.

The Projection Matrix

The Projection Matrix

- We need some way of getting from the set Ω_c to $\mathbb{R}^n$.
- We do this by projecting Ω_c into $\mathbb{R}^n$.

The Projection Matrix

The Projection Matrix

- We need some way of getting from the set Ω_c to $\mathbb{R}^n$.
- We do this by projecting Ω_c into $\mathbb{R}^n$.
- We use the following matrix for this projection:

$$P = I - \nabla\phi \otimes \nabla\phi.$$

The Projection Matrix

The Projection Matrix

- We need some way of getting from the set Ω_c to $\mathbb{R}^n$.
- We do this by projecting Ω_c into $\mathbb{R}^n$.
- We use the following matrix for this projection:

$$P = I - \nabla\phi \otimes \nabla\phi.$$

where " $\otimes$ " denotes the Kronecker product.

The Projection Matrix

The Projection Matrix

- We need some way of getting from the set Ω_c to $\mathbb{R}^n$.
- We do this by projecting Ω_c into $\mathbb{R}^n$.
- We use the following matrix for this projection:

$$P = I - \nabla\phi \otimes \nabla\phi.$$

  where " $\otimes$ " denotes the Kronecker product.
- This matrix projects the system into $\mathbb{R}^n$!

Extension of initial data

- Now we need a way of extending the intial data off of the surface.

The Closest Point Method

The Closest Point Method

The Closest Point Method

The Closest Point Method

1. Compute the signed distance function (some notion of distance on the surface).

The Closest Point Method

The Closest Point Method

1. Compute the signed distance function (some notion of distance on the surface).
2. Choose the computational domain of the form $\Omega_c = x : |\phi(x)| \leq c$.

The Closest Point Method

The Closest Point Method

1. Compute the signed distance function (some notion of distance on the surface).
2. Choose the computational domain of the form $\Omega_c = x : |\phi(x)| \leq c$.
3. Derive the Eulerian representation of the surface PDE.

The Closest Point Method

The Closest Point Method

1. Compute the signed distance function (some notion of distance on the surface).
2. Choose the computational domain of the form $\Omega_c = x : |\phi(x)| \leq c$.
3. Derive the Eulerian representation of the surface PDE.
4. Extend the data off the surface by requiring $\nabla u_0 \cdot \nabla \phi = 0$ for $x \in \Omega_c$

The Closest Point Method

The Closest Point Method

1. Compute the signed distance function (some notion of distance on the surface).
2. Choose the computational domain of the form $\Omega_c = x : |\phi(x)| \leq c$.
3. Derive the Eulerian representation of the surface PDE.
4. Extend the data off the surface by requiring $\nabla u_0 \cdot \nabla \phi = 0$ for $x \in \Omega_c$
5. Compute the Eulerian representation of the surface PDE using standard finite differences on a Cartesian mesh in the computational domain.

Benefits of the CP Method

Benefits of the CP Method

- We don't need a conformal map into $\mathbb{R}^n$ or a triangulation of the surface.

Benefits of the CP Method

- We don't need a conformal map into $\mathbb{R}^n$ or a triangulation of the surface.
- We don't need artificial boundry conditions, nor do we introduce artificial singularities.

Benefits of the CP Method

- We don't need a conformal map into $\mathbb{R}^n$ or a triangulation of the surface.
- We don't need artificial boundry conditions, nor do we introduce artificial singularities.
- This approach allows us to discretize differential operators in Euclidean space.

Benefits of the CP Method

- We don't need a conformal map into $\mathbb{R}^n$ or a triangulation of the surface.
- We don't need artificial boundry conditions, nor do we introduce artificial singularities.
- This approach allows us to discretize differential operators in Euclidean space.
- We can use our typical numerical schemes from Math 164!

Future Work

Future Work

- In the future we hope to be able to:

Future Work

- In the future we hope to be able to:
  - Model evolving surfaces (e.g., motion by mean curvature)

Future Work

- In the future we hope to be able to:
  - Model evolving surfaces (e.g., motion by mean curvature)
  - Solve a PDE on the evolving surface (e.g., heat/reaction-diffusion equations)