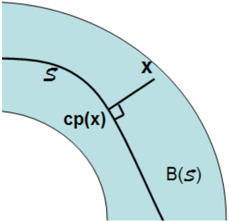
Closest Point Method: Method of Lines Approach

Yujia Chen,
joint work with Colin Macdonald (supervisor),
Ingrid von Glehn, and Tom März

Mathematical Institute, University of Oxford

Oriel CBL course, Oxford, Aug 2014

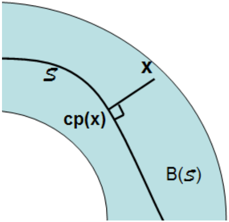
Review of closest point extension and principles



- Given a smooth surface S, let $\mathrm{cp}(x)$ be the closest point on $\mathcal{S}$ to the point x.
- Given $u : \mathcal{S}$ or $B(\mathcal{S}) \to \mathbb{R}$, define the closest point extension: $v(x) := u(\mathrm{cp}(x))$ with $v : B(\mathcal{S}) \to \mathbb{R}$.
- In operator form: $v = Eu$.

Review of closest point extension and principles

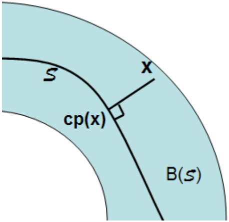
- Given a smooth surface S, let $\text{cp}(x)$ be the closest point on $\mathcal{S}$ to the point x.
- Given $u : \mathcal{S}$ or $B(\mathcal{S}) \to \mathbb{R}$, define the closest point extension: $v(x) := u(\text{cp}(x))$ with $v : B(\mathcal{S}) \to \mathbb{R}$.
- In operator form: $v = Eu$.



Remark: for $w : B(\mathcal{S}) \to \mathbb{R}$, $w = Ew$ simply means that w is constant along the normal to $\mathcal{S}$, and vice versa.

Review of closest point extension and principles

- Given a smooth surface S, let $\mathrm{cp}(x)$ be the closest point on $\mathcal{S}$ to the point x.
- Given $u : \mathcal{S}$ or $B(\mathcal{S}) \to \mathbb{R}$, define the closest point extension: $v(x) := u(\mathrm{cp}(x))$ with $v : B(\mathcal{S}) \to \mathbb{R}$.
- In operator form: $v = Eu$.



Remark: for $w : B(\mathcal{S}) \to \mathbb{R}$, $w = Ew$ simply means that w is constant along the normal to $\mathcal{S}$, and vice versa.

Closest Point Principles

$$\nabla[(Eu)(y)] = \nabla_S u(y), \quad y \in S. \qquad \nabla \cdot [(E\mathbf{A})(y)] = \nabla_S \cdot \mathbf{A}(y), \quad y \in S.$$

$$\Delta[(Eu)(y)] = \Delta_{\mathcal{S}} u(y), \quad y \in \mathcal{S}.$$

Review of Ruuth–Merriman approach: diffusion on circle

Aim to solve $u_t = \Delta_{\mathcal{S}} u$, $u(0, \mathbf{x}) = u_0(\mathbf{x})$.
Extend u from $\mathcal{S}$ to v defined in $B(\mathcal{S})$. Initially, let $v_0 = Eu_0$.

Semi-discretization in time direction:

$$\tilde{v}(t^{n+1}, \mathbf{y}) = v(t^n, \mathbf{y}) + \tau \Delta v(t^n, \mathbf{y})$$
$$v(t^{n+1}, \mathbf{y}) = \tilde{v}(t^{n+1}, \mathrm{cp}(\mathbf{y}))$$

Full discretization:

$$\tilde{v}_{j,k}^{n+1} = v_{j,k}^n + \frac{\tau}{h^2}(v_{j-1,k}^n + v_{j+1,k}^n + v_{j,k-1}^n + v_{j,k+1}^n - 4v_{j,k}^n)$$
$$v_{j,k}^{n+1} = \sum_{(x_l, y_m) \in \mathrm{Interp}(\mathrm{cp}(x_j, y_k))} \omega_{l,m} \tilde{v}_{l,m}^{n+1}$$

Matrix formulation:

$$\tilde{\mathbf{v}}_h^{n+1} = \mathbf{v}_h^n + \tau \mathbf{L}_h \mathbf{v}_h^n$$
$$\mathbf{v}_h^{n+1} = \mathbf{E}_h \tilde{\mathbf{v}}_h^{n+1}$$

Method of Lines: Motivation and description

- **Motivation**
  - Ruuth–Merriman approach uses explicit time-stepping methods.
    $u_t = \Delta_{\mathcal{S}} u$ requires $\tau = O(h^2)$ for stability,
    $u_t = \Delta^2_{\mathcal{S}} u$ requires $\tau = O(h^4)$ for stability.

Method of Lines: Motivation and description

- **Motivation**
  - Ruuth–Merriman approach uses explicit time-stepping methods.
    $u_t = \Delta_{\mathcal{S}} u$ requires $\tau = O(h^2)$ for stability,
    $u_t = \Delta_{\mathcal{S}}^2 u$ requires $\tau = O(h^4)$ for stability.
  - Implicit time-stepping usually has less severe stability conditions.

Method of Lines: Motivation and description

- **Motivation**
  - Ruuth–Merriman approach uses explicit time-stepping methods.
    $u_t = \Delta_{\mathcal{S}} u$ requires $\tau = O(h^2)$ for stability,
    $u_t = \Delta_{\mathcal{S}}^2 u$ requires $\tau = O(h^4)$ for stability.
  - Implicit time-stepping usually has less severe stability conditions.
  - It is difficult to derive an implicit scheme based on Ruuth–Merriman approach.

Method of Lines: Motivation and description

- **Motivation**
  - Ruuth–Merriman approach uses explicit time-stepping methods.
    $u_t = \Delta_{\mathcal{S}} u$ requires $\tau = O(h^2)$ for stability,
    $u_t = \Delta_{\mathcal{S}}^2 u$ requires $\tau = O(h^4)$ for stability.
  - Implicit time-stepping usually has less severe stability conditions.
  - It is difficult to derive an implicit scheme based on Ruuth–Merriman approach.
- **Method of Lines** (MOLs)
  - We first discretize in the spatial direction to get a system of ODEs, then discretize along the time direction using whatever time-stepping schemes we like (explicit, implicit, or semi-implicit).

Method of Lines: Motivation and description

- **Motivation**
  - Ruuth–Merriman approach uses explicit time-stepping methods.
    $u_t = \Delta_{\mathcal{S}} u$ requires $\tau = O(h^2)$ for stability,
    $u_t = \Delta_{\mathcal{S}}^2 u$ requires $\tau = O(h^4)$ for stability.
  - Implicit time-stepping usually has less severe stability conditions.
  - It is difficult to derive an implicit scheme based on Ruuth–Merriman approach.
- **Method of Lines** (MOLs)
  - We first discretize in the spatial direction to get a system of ODEs, then discretize along the time direction using whatever time-stepping schemes we like (explicit, implicit, or semi-implicit).
  - For example, consider the normal heat equation $u_t = \Delta u$ in 2D. We first discretize (spatially) Δu, and get the ODE system $\frac{\partial \mathbf{u}_h}{\partial t} = \mathbf{L}_h \mathbf{u}_h$, where $\mathbf{u}_h$ is a column vector and $\mathbf{L}_h$ is the Laplacian matrix. Then we can use our favorite ODE solvers to solve the ODE system.

MOLs: extending the equation — diffusion on circle

A surface PDE

$$u_t = \Delta_{\mathcal{S}} u$$

Extend both sides of equation

$$E\left(u_t\right) = E\left(\Delta_{\mathcal{S}} u\right)$$

Apply closest point principles $\quad(\Delta_{\mathcal{S}} u = \Delta E u$ on surface$)$

$$(Eu)_t = E\big(\Delta E u\big)$$

Couple to a side condition $\;(v = Eu$, sol'n constant in normal direction$)$

$$\begin{aligned} v_t &= E\Delta v, \\ v &= Ev. \end{aligned}$$

MOLs: adding two equations together — diffusion on circle

Problem (C): constrained problem

$$
\begin{aligned}
v_t &= E\Delta v, \\
v &= Ev.
\end{aligned}
$$

Problem (P): penalty problem

$$
\begin{aligned}
v_t &= E\Delta v - \gamma(v - Ev). \\
v_0 &= Ev_0
\end{aligned}
$$

MOLs: adding two equations together — diffusion on circle

Problem (C): constrained problem

$$v_t = E\Delta v,$$
$$v = Ev.$$

Problem (P): penalty problem

$$v_t = E\Delta v - \gamma(v - Ev).$$
$$v_0 = Ev_0$$

Theorem (von Glehn, März, Macdonald 2013): (C)$\Longleftrightarrow$(P) for any γ.

- (C)$\Longrightarrow$(P) is trivial.
- (P)$\Longrightarrow$(C): $v_t = E\Delta v - \gamma(v - Ev)$ (with $v_0 = Ev_0$) $\implies v = Ev$.

MOLs: adding two equations together — diffusion on circle

Problem (C): constrained problem

$$v_t = E\Delta v,$$
$$v = Ev.$$

Problem (P): penalty problem

$$v_t = E\Delta v - \gamma(v - Ev).$$
$$v_0 = Ev_0$$

Theorem (von Glehn, März, Macdonald 2013): (C)$\Longleftrightarrow$(P) for any γ.

- (C)$\Longrightarrow$(P) is trivial.
- (P)$\Longrightarrow$(C): $v_t = E\Delta v - \gamma(v - Ev)$ (with $v_0 = Ev_0$) $\implies v = Ev$.
  Rearranging (P) we get

$$v_t + \gamma v = E(\Delta v + \gamma v),$$

meaning $v_t + \gamma v$ is constant along the normals to $\mathcal{S}$, which in turn gives:

$$v_t + \gamma v = E(v_t + \gamma v) = Ev_t + \gamma Ev.$$
$$(v - Ev)_t = -\gamma(v - Ev)$$

Let $w = v - Ev$, then $w_0 = v_0 - Ev_0 = 0$, and $w_t = -\gamma w$, so $w \equiv 0$.

Intuition of the penalty term $-\gamma(v - Ev)$

Problem (P): $v_t = E\Delta v - \gamma(v - Ev)$

Intuition of the penalty term $-\gamma(v - Ev)$

Problem (P): $v_t = E\Delta v - \gamma(v - Ev)$

- Starting from an initial v satisfying $v = Ev$, with increment $v_t = E\Delta v$ also constant along the normals, v should (in theory) satisfy $v = Ev$ all the time.
- In practice, when doing numerical computing, small errors in normal directions could grow over time.
- Intuition: $-\gamma(v - Ev)$ would damp out errors in normal directions.

Intuition of the penalty term $-\gamma(v - Ev)$

Problem (P): $v_t = E\Delta v - \gamma(v - Ev)$

- Starting from an initial v satisfying $v = Ev$, with increment $v_t = E\Delta v$ also constant along the normals, v should (in theory) satisfy $v = Ev$ all the time.
- In practice, when doing numerical computing, small errors in normal directions could grow over time.
- Intuition: $-\gamma(v - Ev)$ would damp out errors in normal directions.

Some comments on γ:

- Choice of γ is crucial for numerical computing.
- In practice we want big γ, but not so big to avoid making the penalized problem more difficult to solve than the original problem. For instance, when solving the diffusion equation, we set $\gamma = O(1/h^2)$.
- γ affects consistency and stability of our scheme.

MOLs: general surface PDEs

(von Glehn, März, Macdonald, 2013)

For a general surface PDE: $u_t = f(u, \nabla_{\mathcal{S}}, \nabla_{\mathcal{S}} \cdot, \Delta_{\mathcal{S}})$.
The constrained embedding problem (C):

$$\begin{aligned} v_t &= E\, f(v, \nabla, \nabla \cdot, \Delta), \\ v &= Ev. \end{aligned}$$

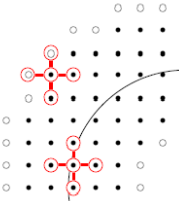
Adding two equations together we obtain the penalty problem (P):

$$\begin{aligned} v_t &= E\, f(v, \nabla, \nabla \cdot, \Delta) - \gamma(v - Ev), \\ v_0 &= Ev_0. \end{aligned}$$

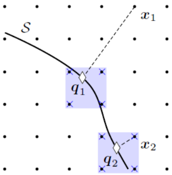
Again problem (C) is equivalent to problem (P) for any γ.

Spatial discretization: back to diffusion example

Standard finite differences:
$\Delta v \to \mathbf{L}^h \mathbf{v}^h$
($\mathbf{L}^h$: Laplacian matrix)

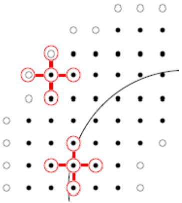


Interpolation schemes:
$Ev \to \mathbf{E}^h \mathbf{v}^h$
($\mathbf{E}^h$: interpolation matrix)

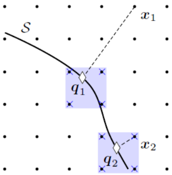


Spatial discretization: back to diffusion example

Standard finite differences:
$\Delta v \to \mathbf{L}^h \mathbf{v}^h$
($\mathbf{L}^h$: Laplacian matrix)



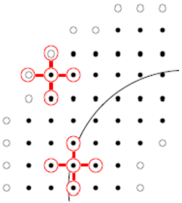
Interpolation schemes:
$Ev \to \mathbf{E}^h \mathbf{v}^h$
($\mathbf{E}^h$: interpolation matrix)



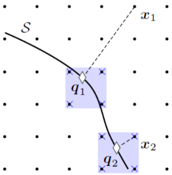
$$v_t = E\Delta v - \gamma(v - Ev)$$

Spatial discretization: back to diffusion example

Standard finite differences:
$\Delta v \to \mathbf{L}^h \mathbf{v}^h$
($\mathbf{L}^h$: Laplacian matrix)



Interpolation schemes:
$Ev \to \mathbf{E}^h \mathbf{v}^h$
($\mathbf{E}^h$: interpolation matrix)



$$v_t = E \Delta v - \gamma (v - Ev)$$

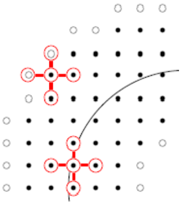
$$\mathbf{v}_t^h = \mathbf{E}_p^h \mathbf{L}^h \mathbf{v}^h - \gamma (\mathbf{I}^h - \mathbf{E}_q^h) \mathbf{v}^h$$

Spatial discretization: back to diffusion example

Standard finite differences:

$$\Delta v \to \mathbf{L}^h \mathbf{v}^h$$

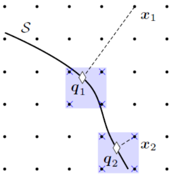
($\mathbf{L}^h$: Laplacian matrix)



Interpolation schemes:

$$Ev \to \mathbf{E}^h \mathbf{v}^h$$

($\mathbf{E}^h$: interpolation matrix)



$$v_t = E\Delta v - \gamma(v - Ev)$$

$$\mathbf{v}^h_t = \mathbf{E}^h_p \mathbf{L}^h \mathbf{v}^h - \gamma(\mathbf{I}^h - \mathbf{E}^h_q)\mathbf{v}^h$$

$$\boxed{\mathbf{v}^h_t = \mathbf{A}^h \mathbf{v}^h, \quad \mathbf{A}^h = \mathbf{E}^h_p \mathbf{L}^h - \gamma(\mathbf{I}^h - \mathbf{E}^h_q)}$$

Several time-stepping methods

Now we are left with an ODE system: $\mathbf{v}_t = \mathbf{A}\mathbf{v}$
More generally, we consider the ODE system: $\mathbf{v}_t = \mathbf{f}(t, \mathbf{v})$.

Forward Euler: $\mathbf{v}^{n+1} = \mathbf{v}^n + \tau \mathbf{f}(t^n, \mathbf{v}^n)$
Backward Differentiation Formula (BDF)
BDF1 (Backward Euler): $\mathbf{v}^{n+1} = \mathbf{v}^n + \tau \mathbf{f}(t^{n+1}, \mathbf{v}^{n+1})$
BDF2: $\mathbf{v}^{n+2} - \frac{4}{3}\mathbf{v}^{n+1} + \frac{1}{3}\mathbf{v}^n = \frac{2}{3}\tau \mathbf{f}(t^{n+2}, \mathbf{v}^{n+2})$
BDF3...
Mid-point rule[1]: $\mathbf{v}^{n+1} = \mathbf{v}^n + \frac{\tau}{2}(\mathbf{f}(t^n, \mathbf{v}^n) + \mathbf{f}(t^{n+1}, \mathbf{v}^{n+1}))$

[1] This corresponds to Crank-Nicolson scheme for diffusion equation