Singular quadrature for layer and volume potentials

Ludvig af Klinteberg (Chair)[1]

Computational Tools for PDEs with Complicated Geometries and Interfaces
Day 3, 12 June 2024

[1]Mälardalen University (MDU), Västerås, Sweden

Session plan

Start: 10.00

- Introduction talk (30–40 min), Ludvig af Klinteberg
- Quick break
- Short talks (5–10 min each)
  - Thomas G. Anderson, Rice
  - David Krantz, KTH
  - Zydrunas Gimbutas, NIST
  - Bowei Wu, UMass Lowell
- Group discussion

End: 12.30

Layer Potentials

We represent solutions to BVPs as layer potentials

$$u(\boldsymbol{x}) = \int_\Gamma k(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y}) ds_{\boldsymbol{y}}$$

with k some weakly singular kernel e.g. Laplace 2D SLP $k(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2\pi} \log \|\boldsymbol{x} - \boldsymbol{y}\|$

Density σ is solution to integral equation such as $(I + K)\sigma = f$ on Γ

Layer Potentials

We represent solutions to BVPs as layer potentials

$$u(\boldsymbol{x}) = \int_\Gamma k(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y}) ds_{\boldsymbol{y}}$$

with k some weakly singular kernel e.g. Laplace 2D SLP $k(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2\pi} \log \|\boldsymbol{x} - \boldsymbol{y}\|$

Density σ is solution to integral equation such as $(I + K)\sigma = f$ on Γ

We will need to evaluate $u(\boldsymbol{x})$ in three different regimes:

- **Smooth:** $\boldsymbol{x}$ far away from Γ everything is easy here
- **Singular:** $\boldsymbol{x}$ on Γ typically occurs when solving the integral equation
- **Nearly singular:** $\boldsymbol{x}$ arbitrarily close to Γ surprisingly difficult

Input geometry

Our input is a geometry parametrization $\Gamma = \{\boldsymbol{y}(t) \ : \ t \in D\}$ line or surface

$$u(\boldsymbol{x}) = \int_\Gamma k(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y}) ds_{\boldsymbol{y}} = \int_D k(\boldsymbol{x}, \boldsymbol{y}(t))\sigma(t)|J_{\boldsymbol{y}}(t)| dt$$

note $\sigma(t) = \sigma(\boldsymbol{y}(t))$

Input geometry

Our input is a geometry parametrization $\Gamma = \{\boldsymbol{y}(t) \ : \ t \in D\}$ line or surface

$$u(\boldsymbol{x}) = \int_\Gamma k(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y}) ds_{\boldsymbol{y}} = \int_D k(\boldsymbol{x}, \boldsymbol{y}(t))\sigma(t)|J_{\boldsymbol{y}}(t)| dt$$

note $\sigma(t) = \sigma(\boldsymbol{y}(t))$

Parametrization can be:

- Analytical
- In terms of basis functions (polynomials, spherical harmonics, ...)
- (Just points will do, because we can interpolate to basis functions)

Layer Potential Discretization

We discretize with a quadrature rule $\{t_j, w_j\}$ for the *smooth* regime,

$$u(\boldsymbol{x}) = \int_\Gamma k(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y}) ds_{\boldsymbol{y}} \approx \sum_{j=1}^{N} k\left(\boldsymbol{x}, \boldsymbol{y}(t_j)\right) \sigma(t_j) |J_{\boldsymbol{y}}(t_j)| w_j,$$

Layer Potential Discretization

We discretize with a quadrature rule $\{t_j, w_j\}$ for the *smooth* regime,

$$u(\boldsymbol{x}) = \int_\Gamma k(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y}) ds_{\boldsymbol{y}} \approx \sum_{j=1}^{N} k\left(\boldsymbol{x}, \boldsymbol{y}(t_j)\right)\sigma(t_j)|J_{\boldsymbol{y}}(t_j)|w_j,$$

- Underlying/smooth quadrature is our base discretization
- Can *interpolate* surface quant. (σ, $\boldsymbol{y}$, $\partial_t \boldsymbol{y}$) from node data

Layer Potential Discretization

We discretize with a quadrature rule $\{t_j, w_j\}$ for the *smooth* regime,

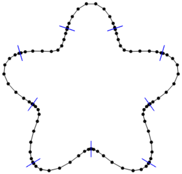
$$u(\boldsymbol{x}) = \int_\Gamma k(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y})ds_{\boldsymbol{y}} \approx \sum_{j=1}^{N} k\left(\boldsymbol{x}, \boldsymbol{y}(t_j)\right)\sigma(t_j)|J_{\boldsymbol{y}}(t_j)|w_j,$$

- Underlying/smooth quadrature is our base discretization
- Can *interpolate* surface quant. (σ, $\boldsymbol{y}$, $\partial_t \boldsymbol{y}$) from node data
- Quadrature either *global* or *local* (composite / panel-based)

Periodic trapezoidal rule (PTR)

trigonometric interpolation

error $\mathcal{O}(e^{-cp})$, $\quad N \sim p^{d-1}$

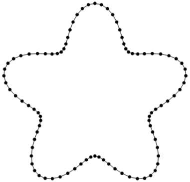


Gauss-Legendre (G-L) panels

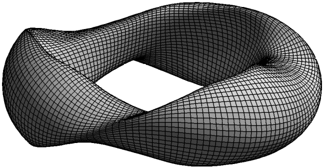
polynomial interpolation

error $\mathcal{O}(h^p)$, $\quad N \sim \left(\frac{p}{h}\right)^{d-1}$

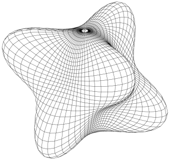
Global and local quadratures



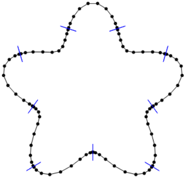
PTR



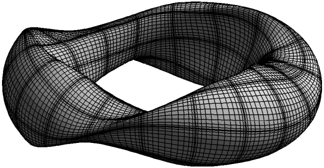
Double PTR



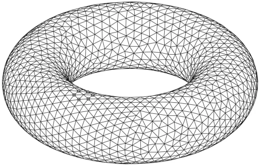
PTR + G-L



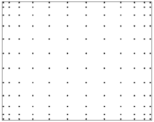
G-L panels

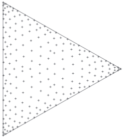


G-L (or Cheb.) quads



Vioreanu-Rokhlin triangles

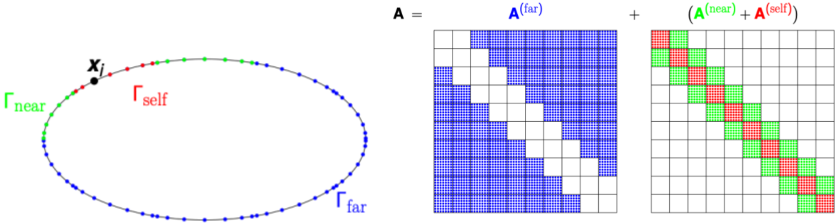




What we need

The linear map that evaluates u from σ at a set of target points $\{\boldsymbol{x}_i\}$, $\boldsymbol{u} = A\boldsymbol{\sigma}$:

$$u(\boldsymbol{x}_i) = \sum_j a_{ij}\sigma_j,$$



What we need

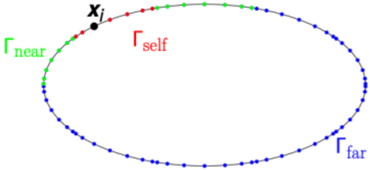
The linear map that evaluates u from σ at a set of target points $\{\boldsymbol{x}_i\}$, $\boldsymbol{u} = A\boldsymbol{\sigma}$:

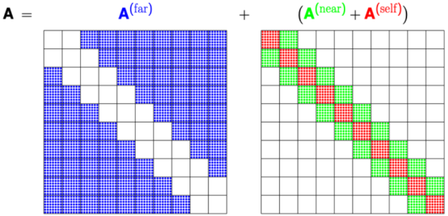
$$u(\boldsymbol{x}_i) = \sum_j a_{ij}\sigma_j,$$

- For targets where smooth quadrature applies,

$$a_{ij} = k\left(\boldsymbol{x}_i, \boldsymbol{y}(t_j)\right)|J_{\boldsymbol{y}}(t_j)|w_j$$

This we can compute $\forall \boldsymbol{x}_i$ using fast methods (FMM)





What we need

The linear map that evaluates u from σ at a set of target points $\{\boldsymbol{x}_i\}$, $\boldsymbol{u} = A\boldsymbol{\sigma}$:

$$u(\boldsymbol{x}_i) = \sum_j a_{ij}\sigma_j,$$

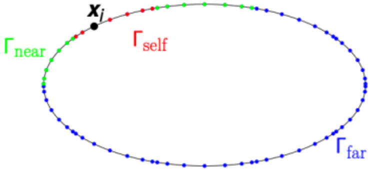
- For targets where smooth quadrature applies,

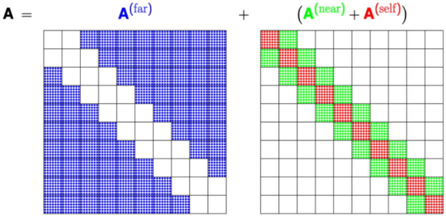
$$a_{ij} = k\left(\boldsymbol{x}_i, \boldsymbol{y}(t_j)\right)|J_{\boldsymbol{y}}(t_j)|w_j$$

  This we can compute $\forall \boldsymbol{x}_i$ using fast methods (FMM)

- For other targets, we need something clever

  Usually target-dependent: not FMM'able.





The problem

The kernel is composed of singularities and smooth parts, e.g.

$$\text{2D: } k(\boldsymbol{x},\boldsymbol{y}) = k_0(\boldsymbol{x},\boldsymbol{y}) + \log\|\boldsymbol{x}-\boldsymbol{y}\| k_L(\boldsymbol{x},\boldsymbol{y}) + \frac{(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{n}}{\|\boldsymbol{x}-\boldsymbol{y}\|^2} k_2(\boldsymbol{x},\boldsymbol{y}) + \dots$$

$$\text{3D: } k(\boldsymbol{x},\boldsymbol{y}) = k_0(\boldsymbol{x},\boldsymbol{y}) + \frac{k_1(\boldsymbol{x},\boldsymbol{y})}{\|\boldsymbol{x}-\boldsymbol{y}\|} + \frac{(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{n}}{\|\boldsymbol{x}-\boldsymbol{y}\|^3} k_3(\boldsymbol{x},\boldsymbol{y}) + \dots$$

with $k_{(\cdot)}(\boldsymbol{x},\boldsymbol{y})$ smooth functions in $\boldsymbol{y}$. We can usually write down this split.

The problem

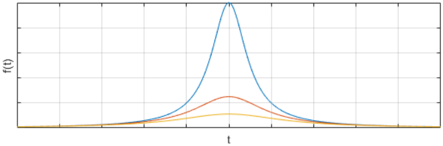
The kernel is composed of singularities and smooth parts, e.g.

$$\text{2D: } k(\boldsymbol{x},\boldsymbol{y}) = k_0(\boldsymbol{x},\boldsymbol{y}) + \log\|\boldsymbol{x}-\boldsymbol{y}\| k_L(\boldsymbol{x},\boldsymbol{y}) + \frac{(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{n}}{\|\boldsymbol{x}-\boldsymbol{y}\|^2} k_2(\boldsymbol{x},\boldsymbol{y}) + \dots$$

$$\text{3D: } k(\boldsymbol{x},\boldsymbol{y}) = k_0(\boldsymbol{x},\boldsymbol{y}) + \frac{k_1(\boldsymbol{x},\boldsymbol{y})}{\|\boldsymbol{x}-\boldsymbol{y}\|} + \frac{(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{n}}{\|\boldsymbol{x}-\boldsymbol{y}\|^3} k_3(\boldsymbol{x},\boldsymbol{y}) + \dots$$

with $k_{(\cdot)}(\boldsymbol{x},\boldsymbol{y})$ smooth functions in $\boldsymbol{y}$. We can usually write down this split.

- **Smooth regime:** $\|\boldsymbol{x}-\boldsymbol{y}\|$ varies slowly when $\boldsymbol{x}$ far from Γ
- **Singular regime:** Weak (integrable) singularities as $\boldsymbol{y}\to\boldsymbol{x}$
- **Nearly singular regime:** $k(\boldsymbol{x},\boldsymbol{y})$ "sharply peaked" when $\boldsymbol{x}$ close to Γ



TODO: Fix better plot

Kernels on Complex Form

Things get simpler in 2D if we use complex variables.
Let (ζ, τ) correspond to $(\boldsymbol{x}, \boldsymbol{y})$ in $\mathbb{C}$, then the Laplace DLP is

$$\int_\Gamma \frac{(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{n}}{\|\boldsymbol{x}-\boldsymbol{y}\|^2}\sigma(\boldsymbol{y})ds_{\boldsymbol{y}} = \operatorname{Im}\int_\Gamma \frac{\sigma(\tau)}{\tau-\zeta}d\tau$$

Kernels on Complex Form

Things get simpler in 2D if we use complex variables.
Let (ζ, τ) correspond to $(\boldsymbol{x}, \boldsymbol{y})$ in $\mathbb{C}$, then the Laplace DLP is

$$\int_\Gamma \frac{(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{n}}{\|\boldsymbol{x}-\boldsymbol{y}\|^2}\sigma(\boldsymbol{y})ds_{\boldsymbol{y}} = \operatorname{Im}\int_\Gamma \frac{\sigma(\tau)}{\tau-\zeta}d\tau$$

Other kernels leave us with integrals on the form

$$\int_\Gamma g(\tau)\log(\tau-\zeta)d\tau \quad \text{and} \quad \int_\Gamma \frac{g(\tau)}{(\tau-\zeta)^q}d\tau, \quad q=1,2,\ldots$$

with $g(\tau)$ smooth and implicitly dependent on ζ.

Kernels on Complex Form

Things get simpler in 2D if we use complex variables.
Let (ζ, τ) correspond to $(\boldsymbol{x}, \boldsymbol{y})$ in $\mathbb{C}$, then the Laplace DLP is

$$\int_\Gamma \frac{(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{n}}{\|\boldsymbol{x}-\boldsymbol{y}\|^2}\sigma(\boldsymbol{y})ds_{\boldsymbol{y}} = \operatorname{Im}\int_\Gamma \frac{\sigma(\tau)}{\tau-\zeta}d\tau$$

Other kernels leave us with integrals on the form

$$\int_\Gamma g(\tau)\log(\tau-\zeta)d\tau \quad \text{and} \quad \int_\Gamma \frac{g(\tau)}{(\tau-\zeta)^q}d\tau, \quad q=1,2,\ldots$$

with $g(\tau)$ smooth and implicitly dependent on ζ.

With parametrization $\Gamma = \{\gamma(t) \,:\, t\in[0,2\pi)\}$

$$\int_\Gamma \frac{g(\tau)}{(\tau-\zeta)^q}d\tau = \int_0^{2\pi}\frac{g(\gamma(t))}{(\gamma(t)-\zeta)^q}|\gamma'(t)|dt = \int_0^{2\pi}\frac{f(t)}{(\gamma(t)-\zeta)^q}dt$$

$f(t) = g(\gamma(t))|\gamma'(t)|$ is also assumed smooth

Singular quadrature: 2D

- Smooth quadrature has fixed (t_j, w_j) for all smooth integrands,

$$\int_a^b f(t)dt = \sum_{j=1}^{n} f(t_i)w_i$$

Singular quadrature: 2D

- Smooth quadrature has fixed (t_j, w_j) for all smooth integrands,

$$\int_a^b f(t)dt = \sum_{j=1}^{n} f(t_i)w_i$$

- Singular quadrature is tailored on type of quadrature and location t_0.
  E.g. $f(t) = k_0(t) + log|t - t_0|k_L(t)$

Singular quadrature: 2D

- Smooth quadrature has fixed (t_j, w_j) for all smooth integrands,

$$\int_a^b f(t)dt = \sum_{j=1}^{n} f(t_i)w_i$$

- Singular quadrature is tailored on type of quadrature and location t_0.
  E.g. $f(t) = k_0(t) + log|t - t_0|k_L(t)$
  - **Product quadrature** integrates the singularity explicitly, $\tilde{w}_i = \tilde{w}_i(t_0)$

$$\int_a^b log|t - t_0|k_L(t)dt = \sum_{j=1}^{n} f_L(t_i)\tilde{w}_i$$

Singular quadrature: 2D

- Smooth quadrature has fixed (t_j, w_j) for all smooth integrands,

$$\int_a^b f(t)dt = \sum_{j=1}^{n} f(t_i)w_i$$

- Singular quadrature is tailored on type of quadrature and location t_0. E.g. $f(t) = k_0(t) + log|t - t_0|k_L(t)$
  - **Product quadrature** integrates the singularity explicitly, $\tilde{w}_i = \tilde{w}_i(t_0)$

$$\int_a^b log|t - t_0|k_L(t)dt = \sum_{j=1}^{n} f_L(t_i)\tilde{w}_i$$

  - **Implicit rules** integrate the complete expression, $\tilde{t}_i, \tilde{w}_i = \tilde{t}_i(t_0), \tilde{w}_i(t_0)$

$$\text{Modified weights: } \int_a^b f(t)dt = \sum_{j=1}^{n} f(t_i)\tilde{w}_i$$

$$\text{Auxiliary nodes: } \int_a^b f(t)dt = \sum_{j=1}^{\tilde{n}} f(\tilde{t}_i)\tilde{w}_i$$

Singular quadrature: 2D Global (PTR)

(Review by Hao et al. 2013)

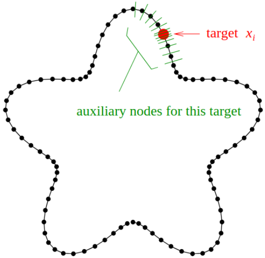
- **Spectral accuracy:**
  Kress product quadrature $\oint_\Gamma \log\|\boldsymbol{x}-\boldsymbol{y}\| f(\boldsymbol{y}) ds_{\boldsymbol{y}} = \sum_{j=1}^{N} f(\boldsymbol{y}_j) w_j$
  Requires split $k_0(\boldsymbol{x},\boldsymbol{y}) + \log\|\boldsymbol{x}-\boldsymbol{y}\| k_L(\boldsymbol{x},\boldsymbol{y})$

Singular quadrature: 2D Global (PTR)

(Review by Hao et al. 2013)

- **Spectral accuracy:**
  Kress product quadrature $\oint_\Gamma \log \|\boldsymbol{x}-\boldsymbol{y}\| f(\boldsymbol{y}) ds_{\boldsymbol{y}} = \sum_{j=1}^{N} f(\boldsymbol{y}_j) w_j$
  Requires split $k_0(\boldsymbol{x},\boldsymbol{y}) + \log \|\boldsymbol{x}-\boldsymbol{y}\| k_L(\boldsymbol{x},\boldsymbol{y})$

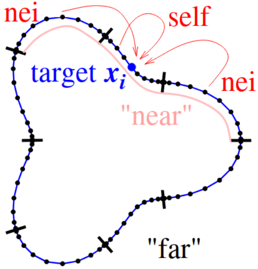
- ***p*th order accuracy:**
  - *Kapur–Rokhlin* modifies fixed number of weights near sing.
  - *Alpert* introduces fixed number of nodes and weights near sing.



Singular quadrature: 2D Local (G-L panels)

Maintains pth order accuracy of panels.

- *Helsing* product quadrature
  - For $\int_{\tau_a}^{\tau_b} g(\tau) \log(\tau - \zeta) d\tau$ and $\int_{\tau_a}^{\tau_b} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau$
  - Requires explicit kernel split

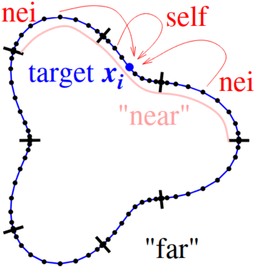


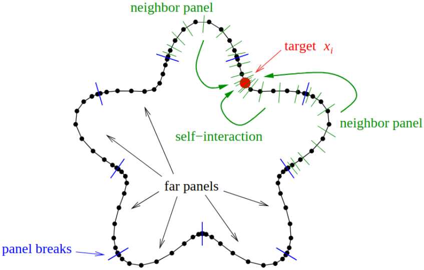
Neighboring panels must be involved.

Singular quadrature: 2D Local (G-L panels)

Maintains *p*th order accuracy of panels.

- *Helsing* product quadrature
  - For $\int_{\tau_a}^{\tau_b} g(\tau) \log(\tau - \zeta) d\tau$ and $\int_{\tau_a}^{\tau_b} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau$
  - Requires explicit kernel split
- *Generalized Gaussian* quadrature
  - Forms nodes and weights through nonlinear optimization
  - Needs singularity type and location, but not explicit split





Neighboring panels must be involved.

Near singularity?

Can understand near singularity through analytic continuation of $\gamma(t)$:

- Find preimage t_0 such that $\gamma(t_0) = \zeta$ Done using poly. approx.

$$\int_D \frac{f(t)dt}{(\gamma(t) - \zeta)^p} = \int_D \frac{f(t)dt}{(\gamma(t) - \gamma(t_0))^p}$$

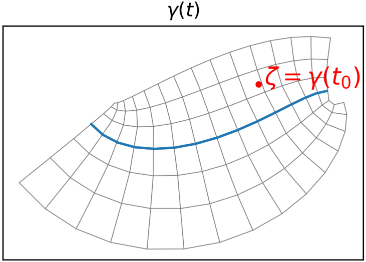
Near singularity?

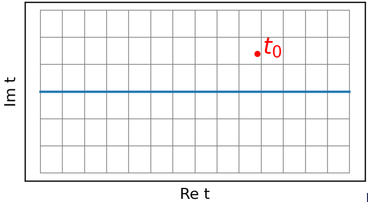
Can understand near singularity through analytic continuation of $\gamma(t)$:

- Find preimage t_0 such that $\gamma(t_0) = \zeta$ Done using poly. approx.

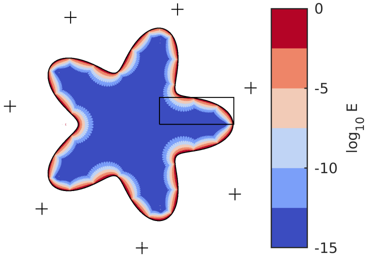
$$\int_D \frac{f(t)dt}{(\gamma(t)-\zeta)^p} = \int_D \frac{f(t)dt}{(\gamma(t)-\gamma(t_0))^p}$$

- Clear where singularity is *in parametrization t*
- t_0 bounds region of analyticity of integrand

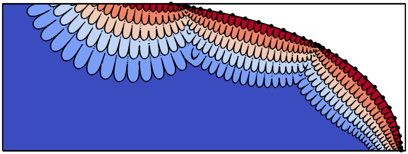




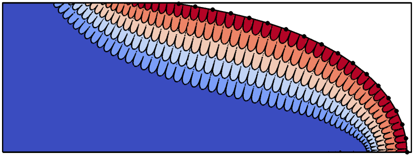
Close evaluation errors



Gauss-Legendre panels



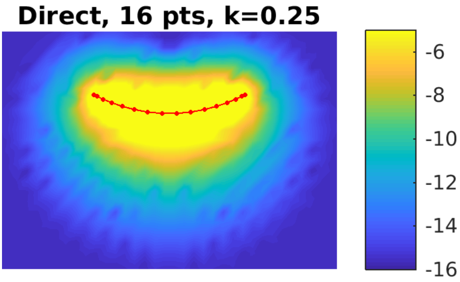
Trapezoidal

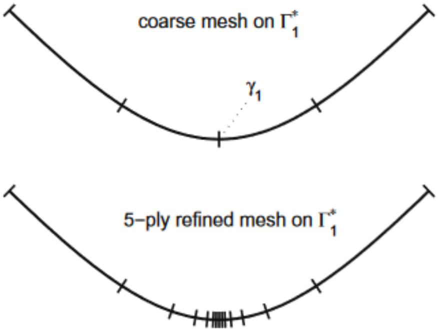


Black contours are estimates based on having found t_0

2D: Nearly singular quadrature

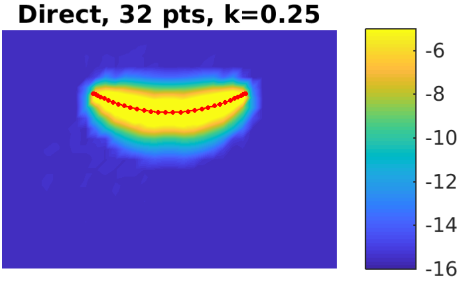
- Many different ideas and methods.
- Easiest: Upsampling to more points
- More efficient: Adaptive refinement near target
- Very effective in 2D: Helsing quadrature (or variant SSQ based on t_0)

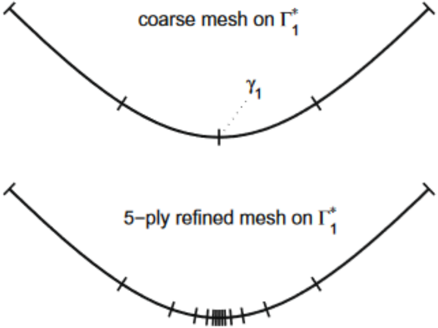




2D: Nearly singular quadrature

- Many different ideas and methods.
- Easiest: Upsampling to more points
- More efficient: Adaptive refinement near target
- Very effective in 2D: Helsing quadrature (or variant SSQ based on t_0)





Interlude: Helsing product quadrature

Want to compute nearly singular

$$I = \int_{\Gamma} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau = \int_{\tau_a}^{\tau_b} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau = \{\text{var. change}\} = \int_{-1}^{1} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau$$

Interlude: Helsing product quadrature

Want to compute nearly singular

$$I = \int_\Gamma \frac{g(\tau)}{(\tau-\zeta)^q} d\tau = \int_{\tau_a}^{\tau_b} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau = \{\text{var. change}\} = \int_{-1}^{1} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau$$

- Form monomial expansion $g(\tau) = \sum_{j=0}^{n-1} c_k \tau^k$ by solving Vandermonde system $V\boldsymbol{c} = \boldsymbol{g}$

Not a problem for $n < 40$

Interlude: Helsing product quadrature

Want to compute nearly singular

$$I = \int_\Gamma \frac{g(\tau)}{(\tau-\zeta)^q} d\tau = \int_{\tau_a}^{\tau_b} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau = \{\text{var. change}\} = \int_{-1}^{1} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau$$

- Form monomial expansion $g(\tau) = \sum_{j=0}^{n-1} c_k \tau^k$ by solving Vandermonde system $V\boldsymbol{c} = \boldsymbol{g}$

  Not a problem for $n < 40$
- Recursively compute $p_k = \int_{-1}^{1} \frac{\tau^k}{(\tau-\zeta)^q} d\tau$

Interlude: Helsing product quadrature

Want to compute nearly singular

$$I = \int_\Gamma \frac{g(\tau)}{(\tau-\zeta)^q} d\tau = \int_{\tau_a}^{\tau_b} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau = \{\text{var. change}\} = \int_{-1}^{1} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau$$

- Form monomial expansion $g(\tau) = \sum_{j=0}^{n-1} c_k \tau^k$ by solving Vandermonde system $V\boldsymbol{c} = \boldsymbol{g}$ Not a problem for $n < 40$
- Recursively compute $p_k = \int_{-1}^{1} \frac{\tau^k}{(\tau-\zeta)^q} d\tau$
- Quadrature is now product

$$\int_{-1}^{1} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau \approx \sum_{j=0}^{n-1} c_k p_k = \boldsymbol{c}^T \boldsymbol{p}$$

Interlude: Helsing product quadrature

Want to compute nearly singular

$$I = \int_\Gamma \frac{g(\tau)}{(\tau-\zeta)^q} d\tau = \int_{\tau_a}^{\tau_b} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau = \{\text{var. change}\} = \int_{-1}^{1} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau$$

- Form monomial expansion $g(\tau) = \sum_{j=0}^{n-1} c_k \tau^k$ by solving Vandermonde system $V\boldsymbol{c} = \boldsymbol{g}$

Not a problem for $n < 40$

- Recursively compute $p_k = \int_{-1}^{1} \frac{\tau^k}{(\tau-\zeta)^q} d\tau$
- Quadrature is now product

$$\int_{-1}^{1} \frac{g(\tau)}{(\tau-\zeta)^q} d\tau \approx \sum_{j=0}^{n-1} c_k p_k = \boldsymbol{c}^T \boldsymbol{p}$$

- We actually want weights w_j s.t. $I \approx \sum g(t_j) w_j = \boldsymbol{g}^T \boldsymbol{w}$
- These we get by solving adjoint Vandermonde $V^T \boldsymbol{w} = \boldsymbol{p}$

Interpolation and precomputation

- Quadrature often ends up needing σ at new nodes (upsamling, adaptive refinement, auxiliary nodes, ...)

$$\int_\Gamma k(\boldsymbol{x},\boldsymbol{y})\sigma(\boldsymbol{y})ds_{\boldsymbol{y}} \approx \sum_{j=1}^{\tilde{n}} k\left(\boldsymbol{x},\boldsymbol{y}(\tilde{t}_j)\right)\sigma(\tilde{t}_j)|J_{\boldsymbol{y}}(\tilde{t}_j)|\tilde{w}_j = \tilde{\boldsymbol{a}}^T\tilde{\boldsymbol{\sigma}},$$

Interpolation and precomputation

- Quadrature often ends up needing σ at new nodes (upsamling, adaptive refinement, auxiliary nodes, ...)

$$\int_\Gamma k(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y}) ds_{\boldsymbol{y}} \approx \sum_{j=1}^{\tilde{n}} k\left(\boldsymbol{x}, \boldsymbol{y}(\tilde{t}_j)\right)\sigma(\tilde{t}_j)|J_{\boldsymbol{y}}(\tilde{t}_j)|\tilde{w}_j = \tilde{\boldsymbol{a}}^T \tilde{\boldsymbol{\sigma}},$$

- Density at new nodes we get through interpolation

$$\tilde{\boldsymbol{\sigma}} = \underbrace{P}_{\tilde{n}\times n}\boldsymbol{\sigma}, \quad \tilde{n} > n$$

Interpolation and precomputation

- Quadrature often ends up needing σ at new nodes (upsamling, adaptive refinement, auxiliary nodes, ...)

$$\int_\Gamma k(\boldsymbol{x},\boldsymbol{y})\sigma(\boldsymbol{y})ds_{\boldsymbol{y}} \approx \sum_{j=1}^{\tilde{n}} k\left(\boldsymbol{x},\boldsymbol{y}(\tilde{t}_j)\right)\sigma(\tilde{t}_j)|J_{\boldsymbol{y}}(\tilde{t}_j)|\tilde{w}_j = \tilde{\boldsymbol{a}}^T\tilde{\boldsymbol{\sigma}},$$

- Density at new nodes we get through interpolation

$$\tilde{\boldsymbol{\sigma}} = \underbrace{P}_{\tilde{n}\times n}\boldsymbol{\sigma}, \quad \tilde{n} > n$$

- If reusable, we can compress this into $\boldsymbol{a} = P^T\tilde{\boldsymbol{a}}$ such that

$$\boldsymbol{a}^T\boldsymbol{\sigma} = \tilde{\boldsymbol{a}}^T\tilde{\boldsymbol{\sigma}}$$

- $\boldsymbol{a}$ corresponds to modified row entries in A matrix

FLATIRON INSTITUTE

QBX

Very different idea: Quadrature By eXpansion. For 2D Laplace DLP

$$u(\zeta) = \int_\Gamma \frac{g(\tau)}{(\tau - \zeta)} d\tau = \sum_{m=0}^{\infty} (\zeta - \zeta_0)^m \underbrace{\int_\Gamma \frac{g(\tau)}{(\tau - \zeta_0)^{m+1}} d\tau}_{c_m(\zeta_0)}$$

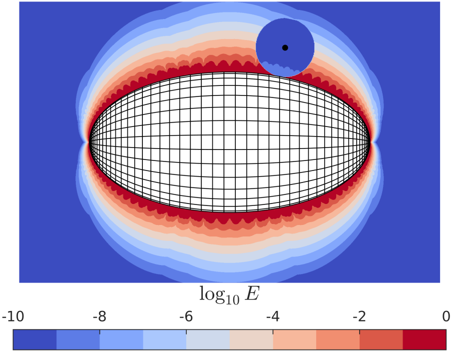
for $|\zeta - \zeta_0| \leq \sup_{\tau \in \Gamma} |\tau - \zeta_0|$.

QBX

Very different idea: Quadrature By eXpansion. For 2D Laplace DLP

$$u(\zeta) = \int_\Gamma \frac{g(\tau)}{(\tau - \zeta)} d\tau = \sum_{m=0}^{\infty} (\zeta - \zeta_0)^m \underbrace{\int_\Gamma \frac{g(\tau)}{(\tau - \zeta_0)^{m+1}} d\tau}_{c_m(\zeta_0)}$$

for $|\zeta - \zeta_0| \leq \sup_{\tau \in \Gamma} |\tau - \zeta_0|$. Same idea extends to 3D.

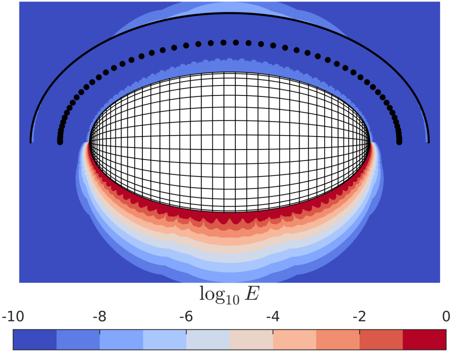


QBX

Very different idea: Quadrature By eXpansion. For 2D Laplace DLP

$$u(\zeta) = \int_\Gamma \frac{g(\tau)}{(\tau - \zeta)} d\tau = \sum_{m=0}^{\infty} (\zeta - \zeta_0)^m \underbrace{\int_\Gamma \frac{g(\tau)}{(\tau - \zeta_0)^{m+1}} d\tau}_{c_m(\zeta_0)}$$

for $|\zeta - \zeta_0| \leq \sup_{\tau \in \Gamma} |\tau - \zeta_0|$. Same idea extends to 3D.

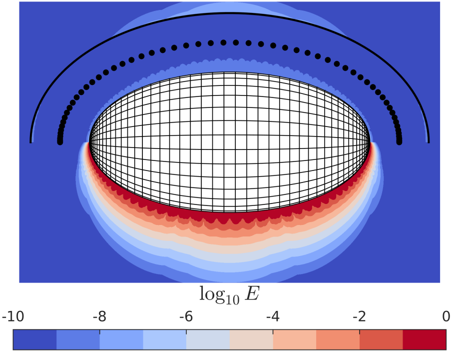


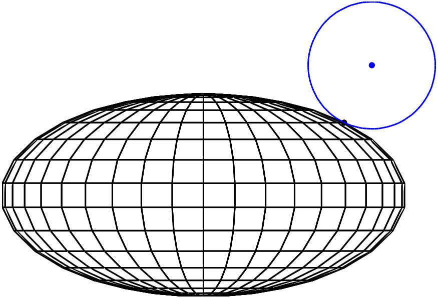
QBX

Very different idea: Quadrature By eXpansion. For 2D Laplace DLP

$$u(\zeta) = \int_\Gamma \frac{g(\tau)}{(\tau - \zeta)} d\tau = \sum_{m=0}^{\infty} (\zeta - \zeta_0)^m \underbrace{\int_\Gamma \frac{g(\tau)}{(\tau - \zeta_0)^{m+1}} d\tau}_{c_m(\zeta_0)}$$

for $|\zeta - \zeta_0| \leq \sup_{\tau \in \Gamma} |\tau - \zeta_0|$. Same idea extends to 3D.



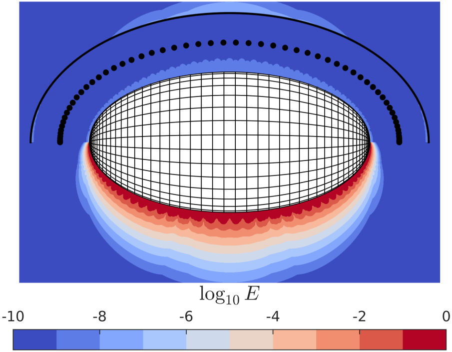


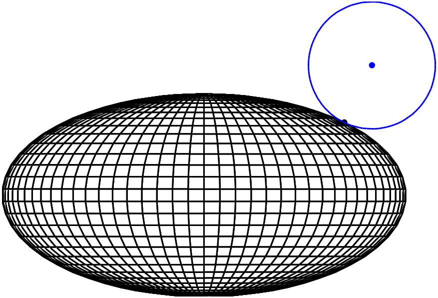
QBX

Very different idea: Quadrature By eXpansion. For 2D Laplace DLP

$$u(\zeta) = \int_\Gamma \frac{g(\tau)}{(\tau - \zeta)} d\tau = \sum_{m=0}^{\infty} (\zeta - \zeta_0)^m \underbrace{\int_\Gamma \frac{g(\tau)}{(\tau - \zeta_0)^{m+1}} d\tau}_{c_m(\zeta_0)}$$

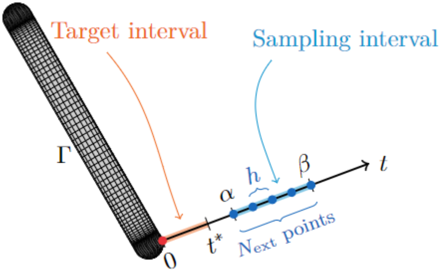
for $|\zeta - \zeta_0| \leq \sup_{\tau \in \Gamma} |\tau - \zeta_0|$. Same idea extends to 3D.





Hedgehog

- Same underlying idea as QBX: Expand solution in domain
- Here: Along a line
- Simple, but hard to optimize



3D: Everything harder

Weak sing. integrable if at center of polar coord. system on surface

$$\int_D k(\boldsymbol{x}, \boldsymbol{y}(r,\theta))\sigma(r,\theta)|J_{\boldsymbol{y}}(r,\theta)|r dr d\theta$$

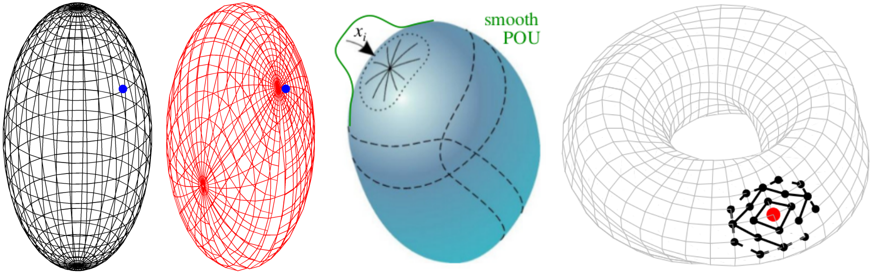
3D: Everything harder

Weak sing. integrable if at center of polar coord. system on surface

$$\int_D k(\boldsymbol{x}, \boldsymbol{y}(r,\theta))\sigma(r,\theta)|J_{\boldsymbol{y}}(r,\theta)|rdrd\theta$$

Global discretization:

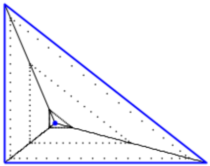
Grid rotations / Local patch / High-order correction



Other approaches: Regularization, singularity subtraction (limited order)

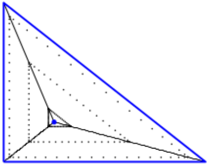
Local discretization: As used in FMM3DBIE

Singular: Auxiliary nodes from generalized Gaussian quadrature

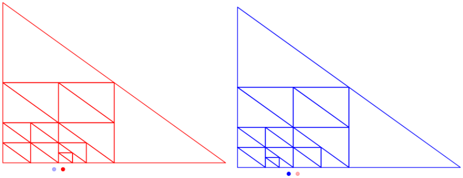


Local discretization: As used in FMM3DBIE

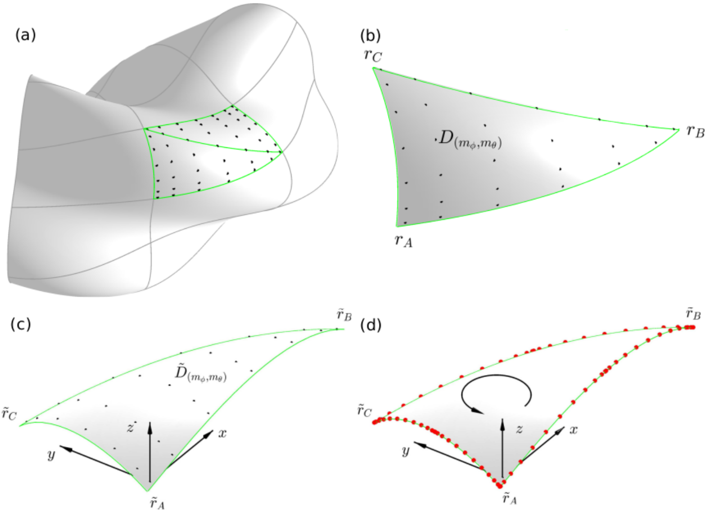
Singular: Auxiliary nodes from generalized Gaussian quadrature



Close eval: Adaptive refinement



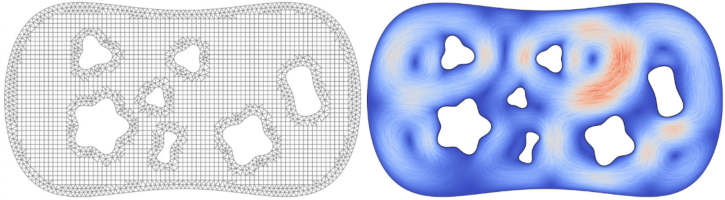
New alternative: Product quadrature using Stokes theorem



Volume potentials

Did not mention these much.

$$V[\sigma](\boldsymbol{x}) = \int_\Omega k(\boldsymbol{x}, \boldsymbol{y})\sigma(\boldsymbol{y}) ds_{\boldsymbol{y}}$$



(Anderson et al.)

Looking forward

- 2D can be considered solved, focus on 3D!
- Wishlist:
  - Close eval & On-surface & Volume
  - High order
  - Fast, parallelizable (moving / deforming geometries)
  - Robust w.r.t. geometry
  - Generalizable to the full zoo of kernels

(Apologies for figures stolen and credit not given.)