An effective discretization scheme for singular integral operators on surfaces

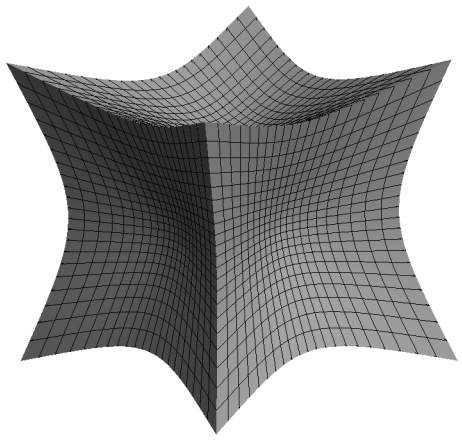
James Bremer

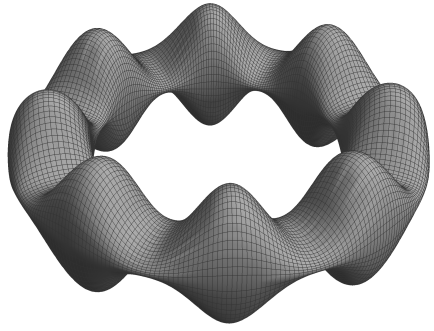
University of Toronto

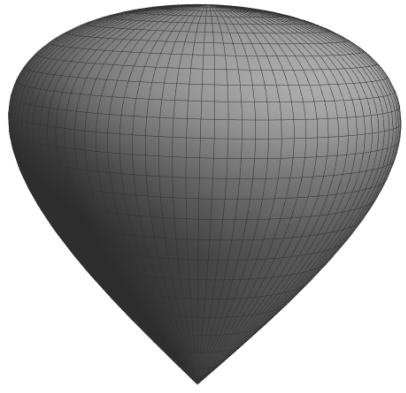
April 9, 2024

Why should we care about singular integral operators on surfaces?

Singular integral operators on surfaces arise when linear elliptic boundary value problems are reformulated as integral equations.







$$\begin{cases} \Delta u(x) + q(x)u(x) = f(x) \quad \text{on} \quad \Omega \\ \dfrac{\partial u}{\partial \nu}(x) = g(x) \quad \text{on} \quad \partial\Omega \end{cases}$$

Global discretization of the differential operator

Perhaps the most obvious way to solve a linear elliptic boundary value problem is via a global spectral scheme.

Many boundary value problems of the form

$$\begin{cases} a^{ij}(x)D_iD_ju(x) + b^i(x)D_iu(x) + c(x)u(x) = f(x) \quad \text{on} \quad \Omega \\ u(x) = g(x) \quad \text{on} \quad \partial\Omega \end{cases}$$

give rise to well-conditioned, invertible operators which act

$$H^2(\Omega) \to L^2(\Omega) \oplus V^{3/2}(\partial\Omega),$$

where $V^{3/2}(\partial\Omega)$ is the trace space of $H^2(\Omega)$.

Advantages: rapidly convergent discretizations which are *as well conditioned as the operator they discretize*

Disadvantages: impractical except on simple geometry; dense discretizations which require elaborate "fast solvers"

Local discretization of the differential operator

Spectral element methods, discontinuous Galerkin methods, finite differences methods and finite element methods can be applied to boundary value problems of the type

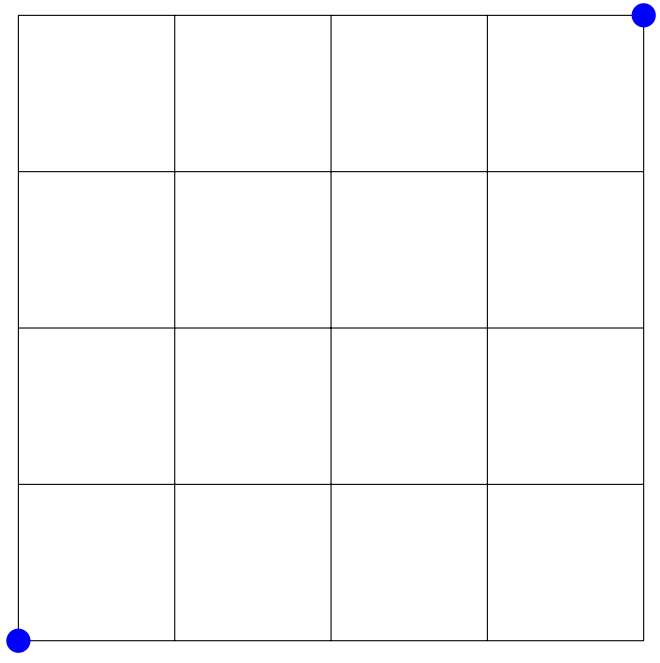
$$\begin{cases} a^{ij}(x)D_iD_ju(x) + b^i(x)D_iu(x) + c(x)u(x) = f(x) \quad \text{on} \quad \Omega \\ u(x) = g(x) \quad \text{on} \quad \partial\Omega. \end{cases}$$

They proceed by subdividing the domain (usually into regular pieces like triangles and tetrahedra) and representing the restriction of the solution to each piece using a local basis (usually polynomials). There are many variations.

Advantages: can handle complicated geometry relatively easily; sparse discretizations; iterative methods are intrinsically "fast"

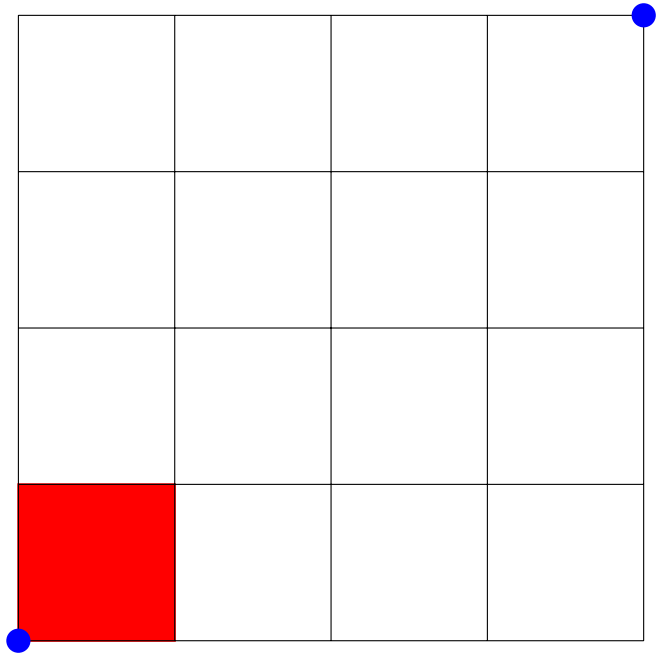
Disadvantages: spatially local discretization of spatially global problem leads to ill-conditioning and other problems

Local discretization of the differential operator



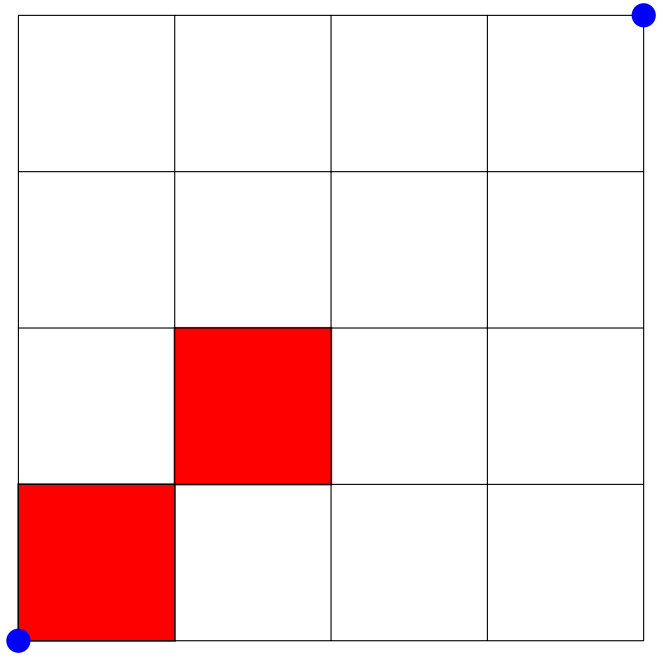
Because of the spatially global nature of the problem, information must travel from one side of the domain to the other.

Local discretization of the differential operator



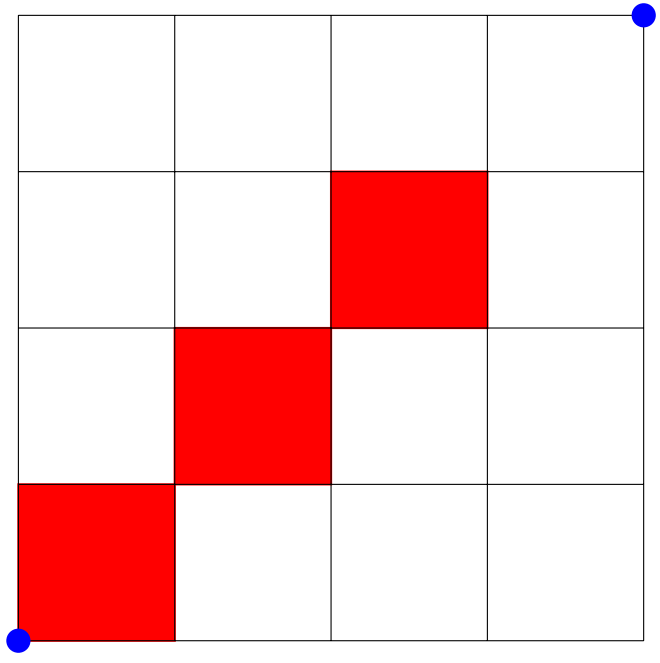
Each application of the discretized operator T only advances information so far ...

Local discretization of the differential operator



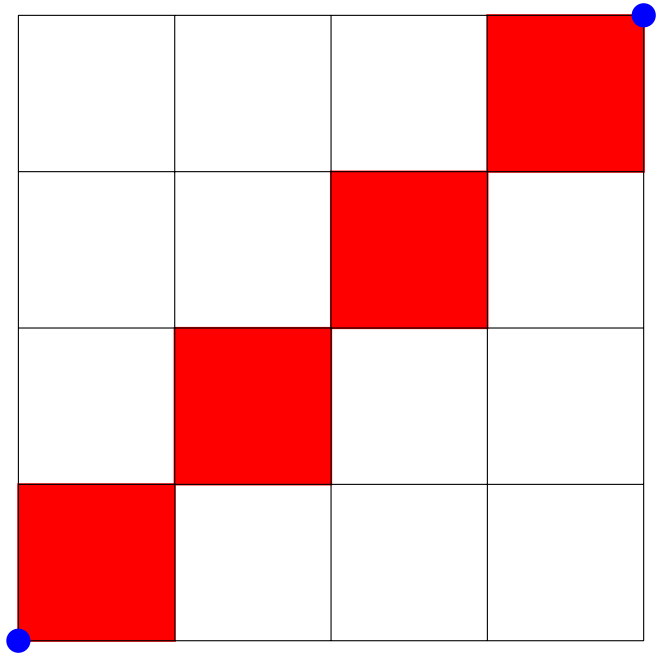
Each application of the discretized operator T only advances information so far ...

Local discretization of the differential operator



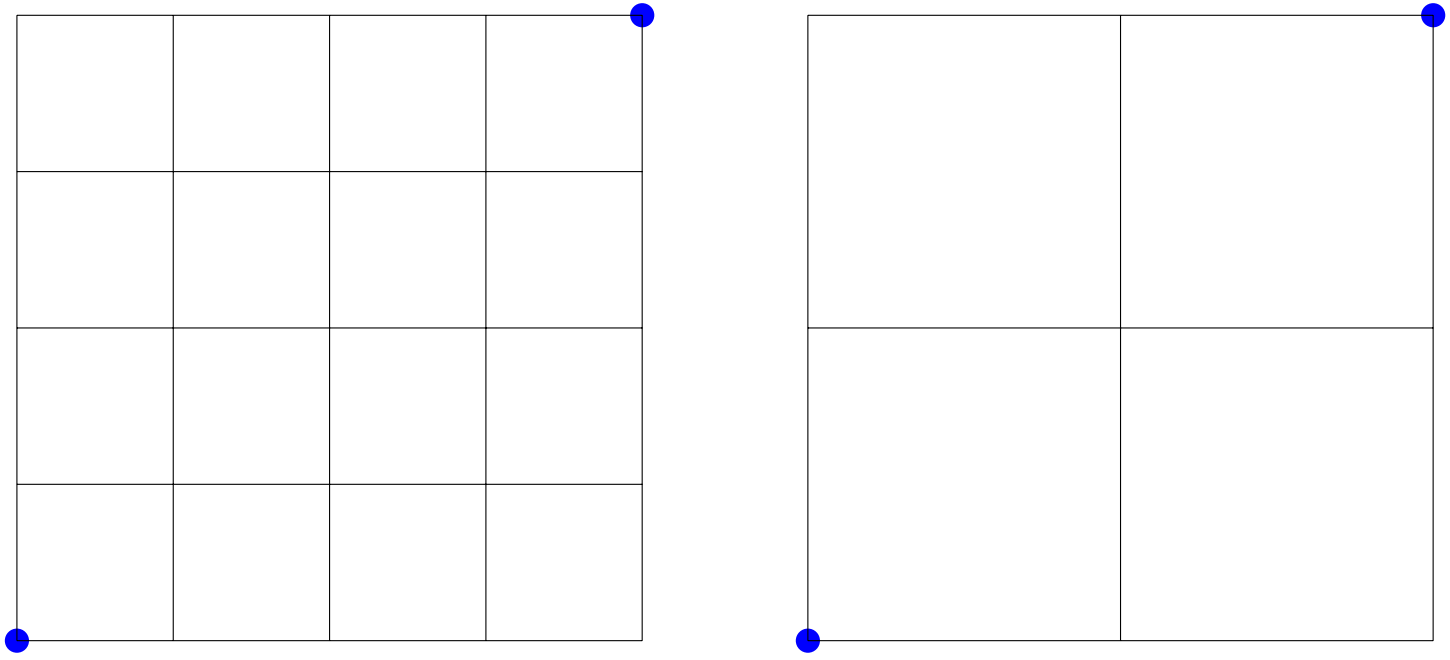
Each application of the discretized operator T only advances information so far ...

Local discretization of the differential operator



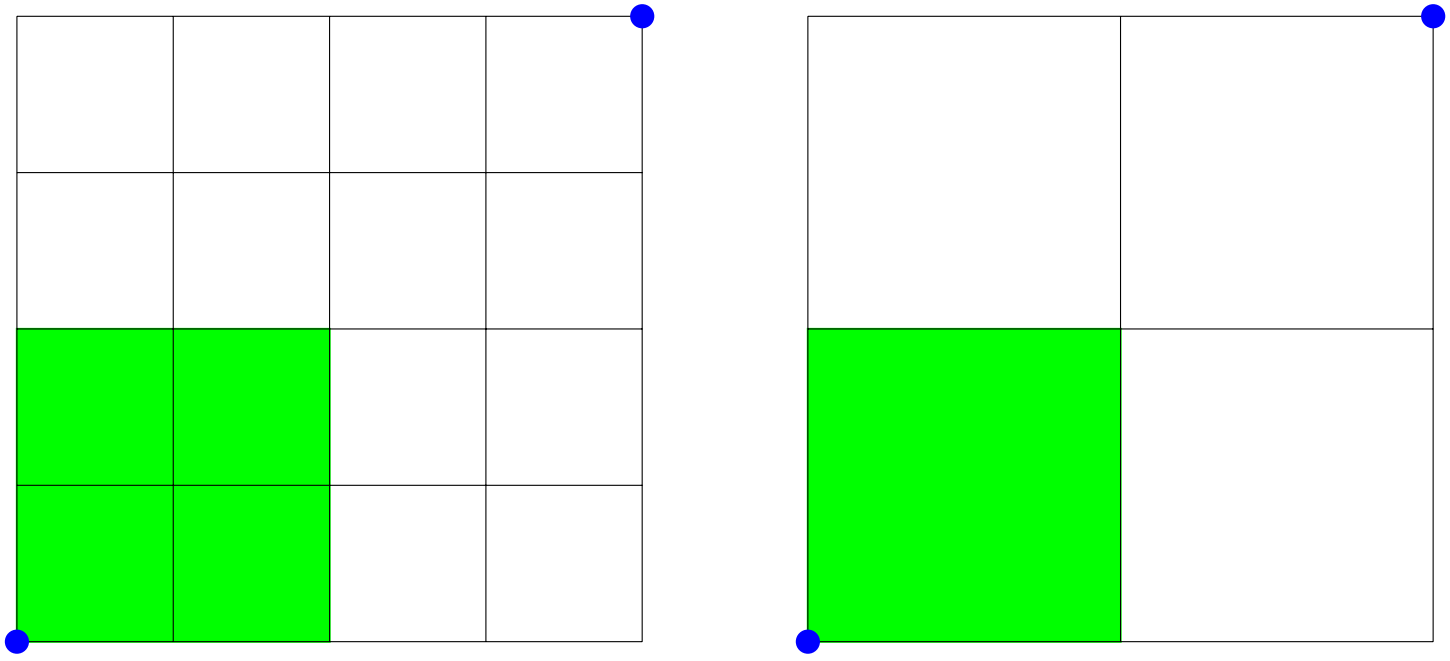
The minimum number of applications we need gives us a bound **from below** on the condition number of the discretization in certain simple cases.

Multigrid methods



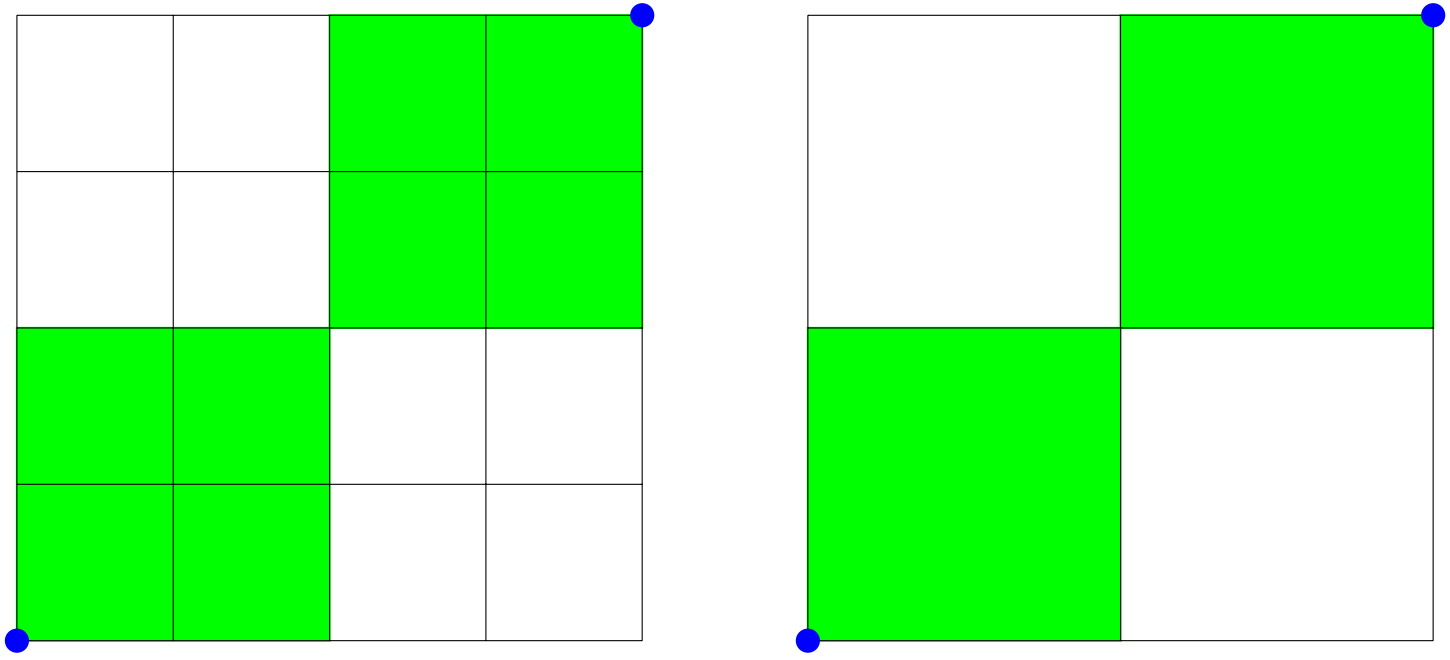
Multigrid methods use multiple grids at different levels of refinement to move information around the domain more efficiently.

Multigrid methods



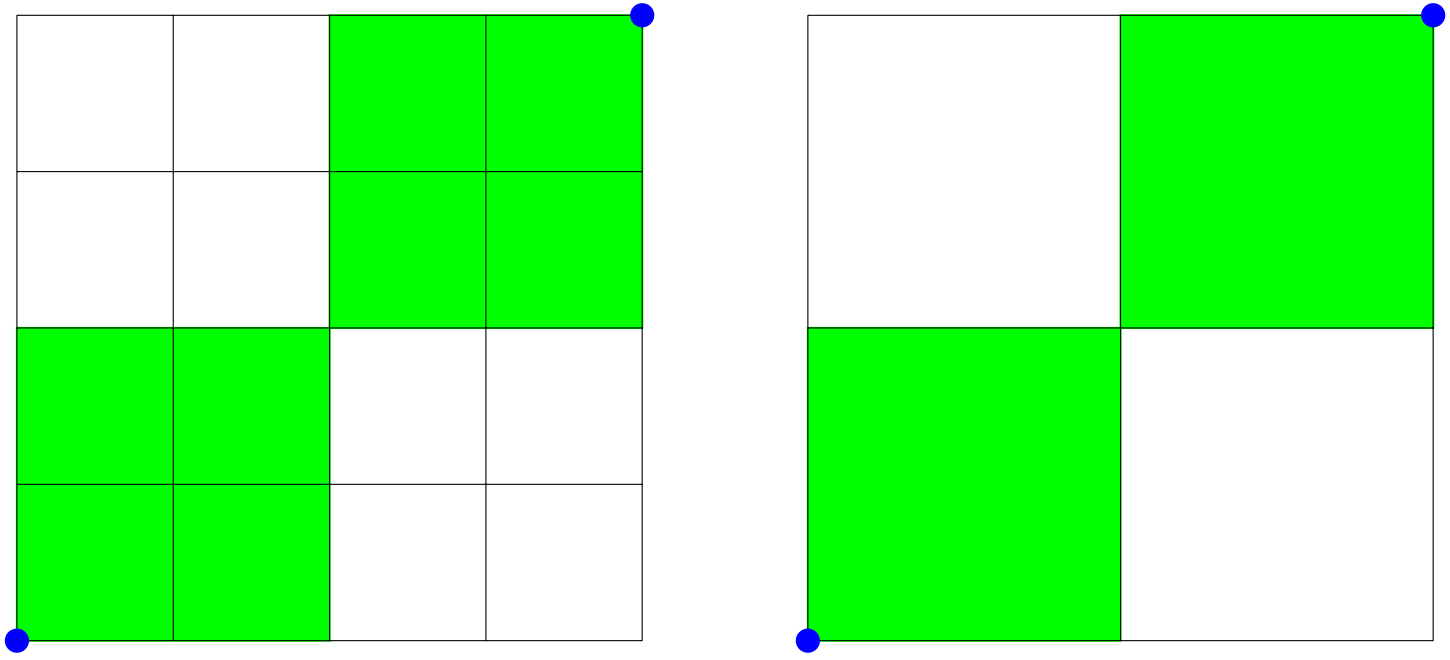
Multigrid methods use multiple grids at different levels of refinement to move information around the domain more efficiently.

Multigrid methods



Multigrid methods use multiple grids at different levels of refinement to move information around the domain more efficiently.

Multigrid methods



Multigrid methods use multiple grids at different levels of refinement to move information around the domain more efficiently.

Advantages: probably the fastest class of methods which achieve reasonable accuracy

Disadvantages: still not as well conditioned as original problem; hard to apply to complicated domains; inelegant

Integral equation methods

Composing the differential operator on the right with a **parametrix** leads to an integral equation.

For instance, we could assume that the solution u of

$$\begin{cases} \Delta u(x) + q(x)u(x) = f(x) & \text{in } \Omega \\ u(x) = g(x) & \text{on } \partial\Omega \end{cases}$$

is of the form

$$u(x) = \int_\Omega K(x,y)\psi(y)\, dy + \int_{\partial\Omega} \tilde{K}(x,y)\sigma(y)\, dS(y)$$

with

$$K(x,y) = \frac{1}{4\pi}\frac{1}{|x-y|} \quad \text{and} \quad \tilde{K}(x,y) = \frac{\partial}{\partial \nu_y} K(x,y) = \frac{1}{4\pi}\frac{(y-x)\cdot \nu_y}{|x-y|^3}.$$

Integral equation methods

This results in a system of integral equations that looks more-or-less like this:

$$\psi(x) + q(x) \int_\Omega K(x,y)\psi(y)\, dy + q(x) \int_{\partial\Omega} \tilde{K}(x,y)\sigma(y)\, dS(y) = f(x) \quad \text{for all} \quad x \in \Omega$$

$$\sigma(x) + q(x) \int_\Omega K(x,y)\psi(y)\, dy + q(x) \int_{\partial\Omega} \tilde{K}(x,y)\sigma(y)\, dS(y) = g(x) \quad \text{for all} \quad x \in \partial\Omega$$

Advantages:

- The operators are bounded $L^2 \to L^2$ and generally well-conditioned
- The unknowns ψ and σ can be represented locally without penalty and without the need match the values of ψ and σ at the artifical boundaries introduced by meshing the domain
- Reduction of dimension in certain cases.

Integral equation methods

This results in a system of integral equations that looks more-or-less like this:

$$\psi(x) + q(x) \int_\Omega K(x,y)\psi(y)\, dy + q(x) \int_{\partial\Omega} \tilde{K}(x,y)\sigma(y)\, dS(y) = f(x) \quad \text{for all} \quad x \in \Omega$$

$$\sigma(x) + q(x) \int_\Omega K(x,y)\psi(y)\, dy + q(x) \int_{\partial\Omega} \tilde{K}(x,y)\sigma(y)\, dS(y) = g(x) \quad \text{for all} \quad x \in \partial\Omega$$

Advantages:

- The operators are bounded $L^2 \to L^2$ and generally well-conditioned
- The unknowns ψ and σ can be represented locally without penalty and without the need match the values of ψ and σ at the artifical boundaries introduced by meshing the domain
- Reduction of dimension in certain cases.

Mathematically very elegant.

Integral equation methods

This results in a system of integral equations that looks more-or-less like this:

$$\psi(x) + q(x) \int_\Omega K(x,y)\psi(y)\, dy + q(x) \int_{\partial\Omega} \tilde{K}(x,y)\sigma(y)\, dS(y) = f(x) \quad \text{for all} \quad x \in \Omega$$

$$\sigma(x) + q(x) \int_\Omega K(x,y)\psi(y)\, dy + q(x) \int_{\partial\Omega} \tilde{K}(x,y)\sigma(y)\, dS(y) = g(x) \quad \text{for all} \quad x \in \partial\Omega$$

Disadvantages:

- Choosing an appropriate parametrix can be difficult.
- Singular integral operators can be hard to discretize.
- The discretized systems are dense.

Integral equation methods

This results in a system of integral equations that looks more-or-less like this:

$$\psi(x) + q(x) \int_\Omega K(x,y)\psi(y)\, dy + q(x) \int_{\partial\Omega} \tilde{K}(x,y)\sigma(y)\, dS(y) = f(x) \quad \text{for all} \quad x \in \Omega$$

$$\sigma(x) + q(x) \int_\Omega K(x,y)\psi(y)\, dy + q(x) \int_{\partial\Omega} \tilde{K}(x,y)\sigma(y)\, dS(y) = g(x) \quad \text{for all} \quad x \in \partial\Omega$$

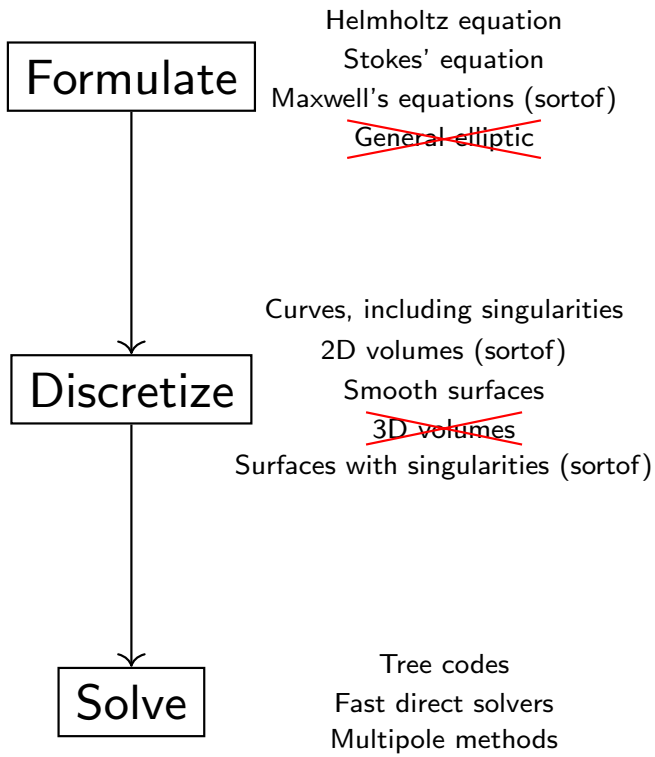
Disadvantages:

- Choosing an appropriate parametrix can be difficult.
- Singular integral operators can be hard to discretize.
- The discretized systems are dense.

Numerical implementation is a nightmare.

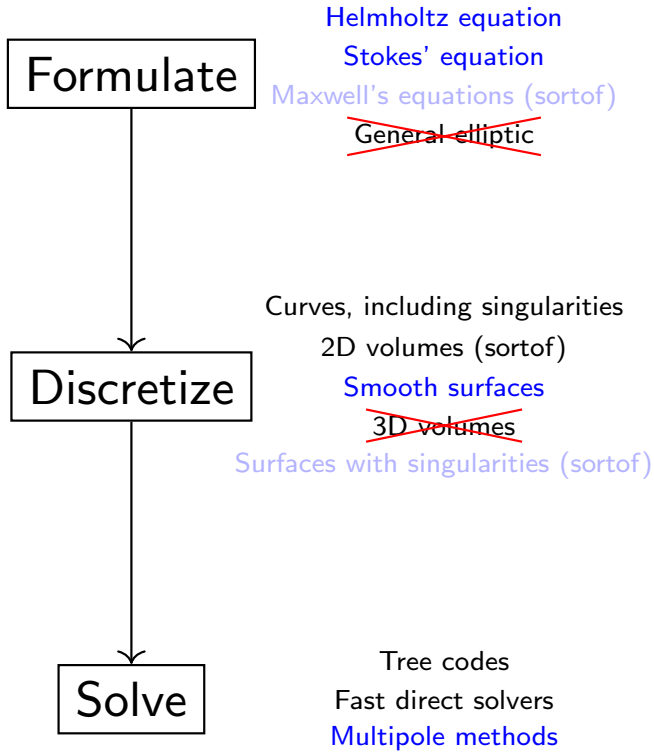
Current state of the art

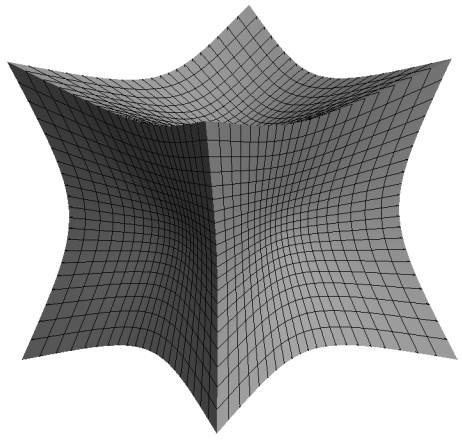
Except in niche cases, integral equation solvers are not competitive with state-of-the-art solvers which use other approaches.



Current state of the art

Recently, an effective package for solving many homogeneous problems was developed by combining fast multipole methods with an approach for discretizing integral equations on surfaces which I developed circa 2012.





$$\begin{cases} \Delta u(x) + k^2 u(x) = 0 & \text{on } \Omega \\ u(x) = g(x) & \text{on } \partial\Omega \end{cases}$$

$$\sigma(x) + D[\sigma](x) = g(x) \qquad (x \in \partial\Omega)$$

$$D[\sigma](x) = \int_{\partial\Omega} \frac{\partial}{\partial \eta_y} \frac{1}{4\pi} \frac{\exp(ik|x-y|)}{|x-y|} \, \sigma(y) \, ds(y)$$

Representation of the surface $\partial\Omega$

We assume that $\partial\Omega$ is described via a collection of smooth mappings given on triangles.

That is, the user provides us with a collection $\{T_j\}$ of triangles in the plane and, for each triangle T_j, also specifies a smooth mapping

$$\rho_j : T_j \to \mathbb{R}^3.$$

- The sets $\rho_j(T_j)$ should form a disjoint cover of $\partial\Omega$.
- This framework includes the case of triangulated surfaces but can incorporate higher order information if it is available.
- Piecewise smooth surfaces can be described with this framework.

Representation of the argument σ

$$D[\sigma](x) = \int_{\partial\Omega} K(x,y)\ \sigma(y)\ ds(y)$$

For each j, we assume that the function

$$\sigma(\rho_j(t))$$

given on the triangle T_j is a polynomial of degree N. We represent this function through its values at the nodes of a quadrature which integrates polynomials of degree $2N$. We call this the discretization quadrature.

To be precise, we represent the restriction of σ to T_j via the vector of scaled values

$$\begin{pmatrix} \sigma(\rho_j(t_1))\sqrt{|d\rho_j(t_1)|}\sqrt{w_1} \\ \sigma(\rho_j(t_2))\sqrt{|d\rho_j(t_2)|}\sqrt{w_2} \\ \vdots \\ \sigma(\rho_j(t_k))\sqrt{|d\rho_j(t_k)|}\sqrt{w_k} \end{pmatrix}$$

where $d\rho_j(t)$ denotes the Jacobian matrix of the parameterization ρ_j at t, $|d\rho_j(t)|$ is the determinant of that matrix and $t_1, \ldots, t_k, w_1, \ldots, w_k$ is the quadrature rule.

Definition of the discretization of D

The matrix which maps

$$\begin{pmatrix} \sigma(\rho_1(t_1))\sqrt{|d\rho_1(t_1)|}\sqrt{w_1} \\ \vdots \\ \sigma(\rho_1(t_m))\sqrt{|d\rho_1(t_m)|}\sqrt{w_m} \\ \hline \sigma(\rho_2(t_1))\sqrt{|d\rho_2(t_1)|}\sqrt{w_1} \\ \vdots \\ \sigma(\rho_2(t_m))\sqrt{|d\rho_2(t_m)|}\sqrt{w_m} \\ \hline \vdots \end{pmatrix} \mapsto \begin{pmatrix} D[\sigma](\rho_1(t_1))\sqrt{|d\rho_1(t_1)|}\sqrt{w_1} \\ \vdots \\ D[\sigma](\rho_1(t_m))\sqrt{|d\rho_1(t_m)|}\sqrt{w_m} \\ \hline D[\sigma](\rho_2(t_1))\sqrt{|d\rho_2(t_1)|}\sqrt{w_1} \\ \vdots \\ D[\sigma](\rho_2(t_m))\sqrt{|d\rho_2(t_m)|}\sqrt{w_m} \\ \hline \vdots \end{pmatrix}$$

Because of our choice of quadrature and the presence of square root weighting, this is a well-behaved Galerkin discretization, and inherits all of the nice properties of such methods (e.g., quasioptimal convergence).

Basic computational unit of the discretizer

A routine which evaluates a block of the discretization matrix corresponding to a collection of target nodes $\xi_1, \ldots, \xi_l$ and a single source triangle T.

Assuming ρ is the parameterization given on T and $w_1, \ldots, w_l$ are the quadrature weights associated with $\xi_1, \ldots, \xi_l$, this block is the $l \times m$ matrix which maps

$$\begin{pmatrix} \sigma(\rho(t_1))\sqrt{d\rho(t_1)}\sqrt{w_1} \\ \vdots \\ \sigma(\rho(t_m))\sqrt{d\rho(t_m)}\sqrt{w_m} \end{pmatrix} \mapsto \begin{pmatrix} \sqrt{w_1}\int_T K(\xi_1, \rho(s))\sigma(\rho(s))\,|d\rho(s)|\,ds \\ \vdots \\ \sqrt{w_l}\int_T K(\xi_l, \rho(s))\sigma(\rho(s))\,|d\rho(s)|\,ds \end{pmatrix}$$

For the sake of simplicity, we will assume there is only one target node ξ with corresponding weight w in what follows.

We form the $1 \times m$ block by first constructing a quadrature rule

$$\int_T K(\xi, \rho(s))\sigma(\rho(s)) \, |d\rho(s)| \, ds \approx \sum_{j=1}^{l} K(\xi, \rho(s_j))\sigma(\rho(u_j)) \, |d\rho(u_j)| \, v_j.$$

Next, we form the vector

$$\begin{pmatrix} K(\xi, \rho(s_1))\sqrt{v_1}\sqrt{w}\sqrt{|d\rho(s_1)|} \\ K(\xi, \rho(s_2))\sqrt{v_2}\sqrt{w}\sqrt{|d\rho(s_2)|} \\ \vdots \\ K(\xi, \rho(s_l))\sqrt{v_l}\sqrt{w}\sqrt{|d\rho(s_l)|} \end{pmatrix}$$

and apply the interpolation matrix which maps

$$\begin{pmatrix} \sigma(\rho(t_1))\sqrt{|d\rho(t_1)|}\sqrt{w_1} \\ \sigma(\rho(t_2))\sqrt{|d\rho(t_2)|}\sqrt{w_2} \\ \vdots \\ \sigma(\rho(t_m))\sqrt{|d\rho(t_m)|}\sqrt{w_m} \end{pmatrix} \rightarrow \begin{pmatrix} \sigma(\rho(s_1))\sqrt{|d\rho(s_1)|}\sqrt{v_1} \\ \sigma(\rho(s_2))\sqrt{|d\rho(s_2)|}\sqrt{v_2} \\ \vdots \\ \sigma(\rho(s_l))\sqrt{|d\rho(s_l)|}\sqrt{v_l} \end{pmatrix}$$

to its right-hand side.

Quadrature rules

All that remains is the construct the appropriate quadrature rules for the integral

$$\int_T K(\xi, \rho(s))\sigma(\rho(s))\,|d\rho(s)|\;\; ds,$$

where

- T is a triangle
- $\rho : T \to \mathbb{R}^3$ is a smooth mapping
- $d\rho(s)$ denotes the Jacobian matrix of ρ at the point s;
- ξ is a point on $\partial\Omega$.

There are three regimes, depending on the location of ξ relative to $\rho(T)$:

- far regime: ξ is "far" from $\rho(T)$.
- nearly singular regime: ξ is "close" to, but not in, $\rho(T)$;
- singular regime: ξ is inside of $\rho(T)$;

Trivial scheme for far-field interactions

When ξ is far from the surface region $\rho(T)$, the integral

$$\int_T K(\xi, \rho(s))\sigma(\rho(s))\,|d\rho(s)|\;\; ds$$

can be evaluated via the discretization quadrature on T used to represent σ. Thus the interpolation matrix is the identity in this case, and the corresponding entries of the discretization matrix are simply of the form

$$K(\xi, \rho(x_j))\sqrt{w}\sqrt{w_j}.$$

Obviously, we do not actually apply any interpolation matrices in this regime.

Thankfully, this is the most commonly occurring case.

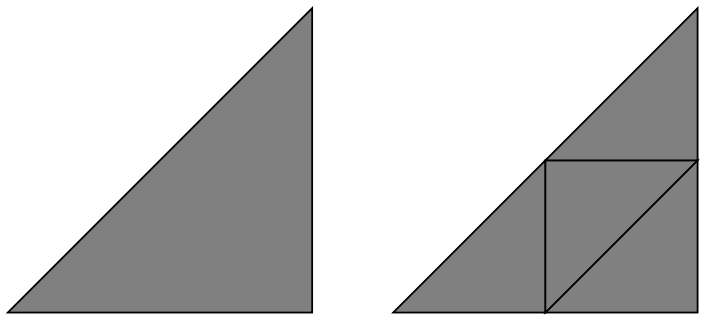
Adaptive scheme for near interactions

In order to evaluate

$$\int_T K(\xi, \rho(y))\sigma(\rho(y))\,|d\rho(y)|$$

with ξ near T, we recursively divide T into a disjoint collection of triangles $T_1, \ldots, T_l$ until the point ξ is in the far-field of every traingle.

We interpolate the values of σ from the discretization quadrature nodes on T to the discretization quadrature nodes on each of the triangles T_j and then evaluate each of the integrals



$$\int_{T_j} K(\xi, \rho(y))\sigma(\rho(y))\,|d\rho(y)|$$

using the discretization quadratures.

This is surprisingly efficient since each division corresponds to the application of the same $4k \times k$ matrix.

It is important to handle multiple target nodes at once during this step.

The singular regime

This is the least frequently occurring regime, but the most challenging. Almost all existing methods fail to accurately evaluate these integrals, except when $\partial\Omega$ is extremely simple.

Changing to polar coordinates takes

$$\int_T K(\xi, \rho(s))\sigma(\rho(s))\,|d\rho(s)|\,dy$$

into an integral of the form

$$\int_0^{2\pi}\int_0^{R(\theta)} \Big(q_{-1}(\theta) + q_0(\theta)r + q_1(\theta)r^2 + q_2(\theta)r^3 + \cdots\Big)\,dr\,d\theta$$

where $R(\theta)$ is the parameterization of the boundary of the integration domain T in polar coordinates and the $q_j(\theta)$ are periodic and analytic in some strip containing the real axis.

Exponential convergence can be obtained by dividing the outer integral into regions on which $R(\theta)$ is smooth and applying tensor product Gaussian quadratures.

Difficulties with this standard approach

The integrand

$$q_{-1}(\theta) + q_0(\theta)r + q_1(\theta)r^2 + q_2(\theta)r^3 + \cdots$$

is analytic on a strip containing the real line but each q_j is of the form

$$q_j(\theta) = \frac{r_j(\theta)}{[I(\theta)]^{j+2}},$$

where r_j is a trigonometric polynomial of **finite order** (which depends on j) but $I(\theta)$ can have zeros close to the real axis.

In fact, if $\{\xi_1, \xi_2\}$ is the basis of the tangent space to the surface at the target node induced by the parameterization ρ, then the zeros of I are the solutions w of the equation

$$\cot(w) = -\lambda^{-1} \exp(\pm i\eta)$$

where

$$\cos(\eta) = \frac{\xi_1 \cdot \xi_2}{|\xi_1|\,|\xi_2|} \quad \text{and} \quad \lambda = \frac{|\xi_1|}{|\xi_2|}.$$

See, for instance, Wendland & Schwab.

Difficulties with this standard approach

The integrand

$$q_{-1}(\theta) + q_0(\theta)r + q_1(\theta)r^2 + q_2(\theta)r^3 + \cdots$$

is analytic on a strip containing the real line but each q_j is of the form

$$q_j(\theta) = \frac{r_j(\theta)}{[I(\theta)]^{j+2}},$$

where r_j is a trigonometric polynomial of **finite order** (which depends on j) but $I(\theta)$ can have zeros close to the real axis.

That is, the proximity of the poles of the functions q_j to the real axis is a measure of how nonconformal the mapping ρ is at the target node ξ. The less conformal (more stretched out), the closer the poles are to the real line.

Difficulties with this standard approach

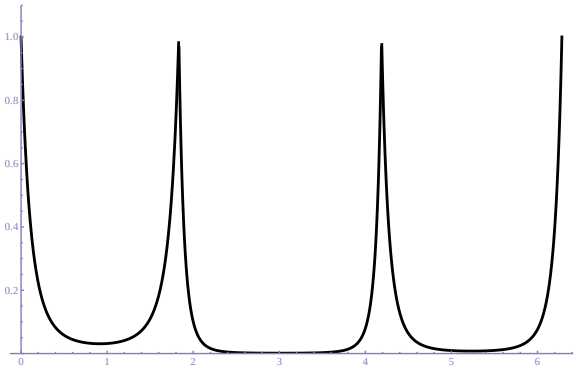
The outer integral in

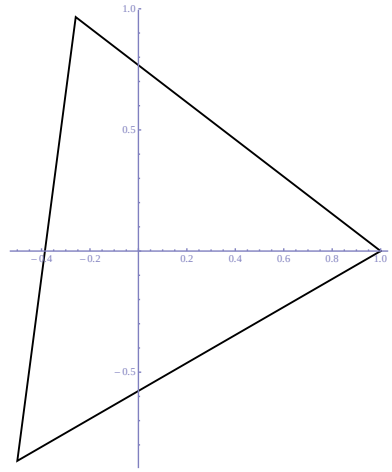
$$\int_0^{2\pi} \int_0^{R(\theta)} \left(q_{-1}(\theta) + q_0(\theta) r + q_1(\theta) r^2 + q_2(\theta) r^3 + \cdots \right) dr\, d\theta$$

is of the form

$$\int_0^{2\pi} \left(q_{-1}(\theta) R(\theta) + q_0(\theta) \frac{(R(\theta))^2}{2} + q_1(\theta) \frac{(R(\theta))^3}{3} + q_2(\theta) \frac{(R(\theta))^4}{4} + \cdots \right) d\theta$$

and the functions $(R(\theta))^k$, while analytic on the real line, are also poorly behaved.





Summary

Exponential convergence is obtained by this standard method. That is, the error in the approximation of the integral obtained using m-point product Legendre quadratures behaves as

$$C \exp\left(-\lambda \cdot m\right).$$

Nonetheless, the method isn't practical because the convergence is slow whenever one of the following conditions applies:

- The mapping ρ is moderately nonconformal at the target node ξ.
- The target node is close to the boundary of the parameterization domain T.
- The parameterization domain T is "stretched out."

Summary

Exponential convergence is obtained by this standard method. That is, the error in the approximation of the integral obtained using m-point product Legendre quadratures behaves as

$$C \exp\left(-\lambda \cdot m\right).$$

Nonetheless, the method isn't practical because the convergence is slow whenever one of the following conditions applies:

- The mapping ρ is moderately nonconformal at the target node ξ.
- The target node is close to the boundary of the parameterization domain T.
- The parameterization domain T is "stretched out."

Standard methods do not work in practice.

Simplification of the integrands

Observation:

Inducing conformality at the target node simplifies the integrand.

Indeed, when the parameterization ρ is conformal at the target node x, the function $I(\theta)$ becomes a constant. In this case, the kernel admits representation as

$$K(\rho(x), \rho(x + (r\cos(\theta), r\sin(\theta)))) = \frac{q_{-1}(\theta)}{r} + q_0(\theta) + q_1(\theta)r + q_2(\theta)r^2 + \cdots$$

where the q_j are **trigonometric polynomials of finite order**.

NOTE: The parameterization ρ only needs to be conformal at one point; it does **not** need be conformal on its whole domain.

Not enough

Inducing conformality at the target node by modifying the parameterization (i.e., by stretching out the triangle T) greatly simplifies the integrand.

But it does nothing to make the evaluation of the outer integral

$$\int_0^{2\pi} \left(q_{-1}(\theta) R(\theta) + q_0(\theta) \frac{R(\theta)^2}{2} + q_1(\theta) \frac{R(\theta)^3}{3} + q_2(\theta) \frac{R(\theta)^4}{4} + \cdots \right) d\theta$$

simpler.

Indeed, it often makes the evaluation of this integral **more difficult.** In many cases, the modified triangle is more stretched out than the original triangle T and this worsens the behavior of the function $R(\theta)$.

Solution: A table of quadrature rules

We precompute table of quadrature rules which allows for the efficient evaluation of the integrals

$$\int_0^{2\pi} \left(q_{-1}(\theta) R(\theta) + q_0(\theta) \frac{R(\theta)^2}{2} + q_1(\theta) \frac{R(\theta)^3}{3} + q_2(\theta) \frac{R(\theta)^4}{4} + \cdots \right) d\theta$$

where R is the parameterization of any triangle — no matter how stretched.

Generalized quadrature

We can construct a quadrature for a collection of user-specified functions

$$f_1(x),\ f_2(x),\ f_3(x), \ldots, f_n(x)$$

on an interval $[a, b]$ by producing a solution $x_1, \ldots, x_m, w_1, \ldots, w_m$ to the nonlinear system of equations

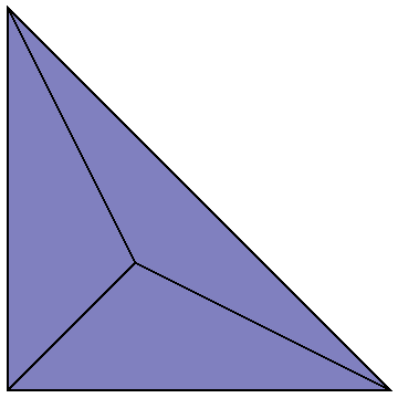
$$\sum_{j=1}^{m} f_i(x_j) w_j = \int_a^b f_i(x) dx, \quad i = 1, \ldots, n.$$

Ideally, m would be equal to $n/2$.

B—, Rokhlin and Gimbutas, "A nonlinear optimization procedure for generalized Gaussian quadratures." *SIAM Journal of Scientific Computing* 32 (2010), pp. 1761-1788.

Quadratures for radially singular functions on triangles

We proceed by dividing the triangle T into three pieces by connecting the target node to the vertices and treating each piece separately.



The effect of this action is to reduce the number of parameters we will need.

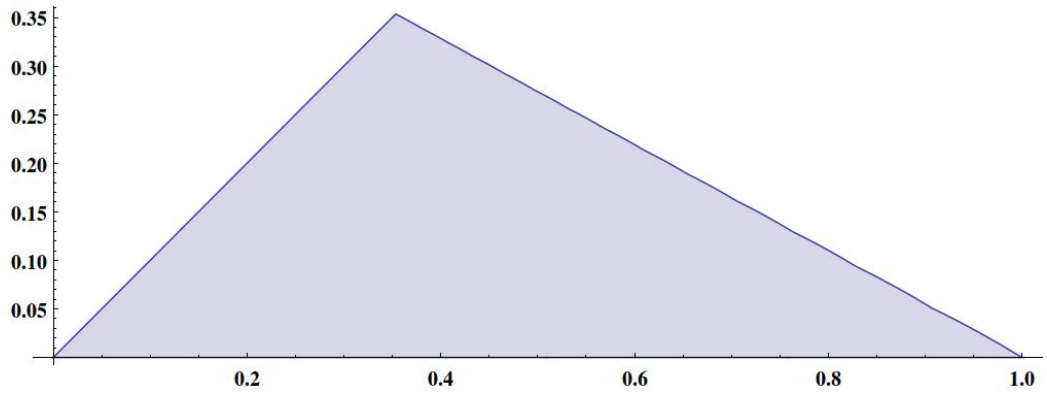
Each of the resulting triangles can be transformed into a triangle with vertices

$$(0,0), \quad (1,0) \quad \text{and} \quad (\alpha\cos(\phi), \alpha\sin(\phi)).$$

without effecting the representation of the kernel underlying this approach.

Quadratures for radially singular functions on triangles

$$R_{\alpha,\phi}(\theta) = \frac{\alpha \sin(\theta)}{\sin(\theta) - \alpha \sin(\theta - \phi)}$$



We wish to evaluate integrals of the form

$$\int_0^{\phi} \int_0^{R_{\alpha,\phi}(\theta)} \left(p_{-1}(\theta) + p_0(\theta) r + p_1(\theta) r^2 + \ldots \right) dr \; d\theta.$$

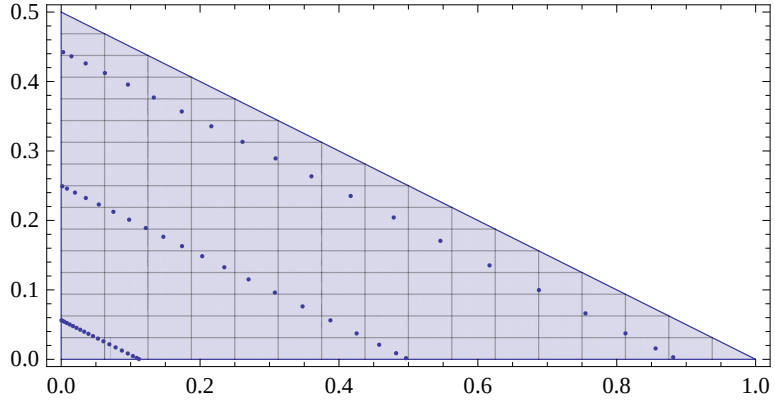
The inner integral can be evaluated efficiently via a Legendre quadrature; so we focus on the outer integral, which can be rewritten in the form

$$\int_0^1 \left(p_{-1}(\phi\theta) R_{\alpha,\phi}(\phi\theta) + p_0(\phi\theta) \frac{R^2_{\alpha,\phi}(\phi\theta)}{2} + \cdots + p_N(\phi\theta) \frac{R^{N+2}_{\alpha,\phi}(\phi\theta)}{N+2} \right) \phi d\theta$$

Each quadrature formula is designed to handle integrals of this type for a fixed range of the parameters α and ϕ and a fixed value of the integer N.

We call N the order of the formula.

$$\iint_{T_\alpha} \frac{\cos(2\theta)}{2r}\, dx\, dy \quad \text{with} \quad T_\alpha = \{(x,y) : 0 \le x \le 1, \quad 0 \le y \le \alpha - \alpha x\}$$



α	N_{adap}	E_{adap}	N_{precomp}	E_{precomp}
$1/2$	854	2.03×10^{-15}	84	2.38×10^{-16}
1×10^{-1}	1078	1.79×10^{-15}	84	1.46×10^{-15}
1×10^{-2}	1302	1.46×10^{-15}	84	8.34×10^{-16}
1×10^{-3}	1526	4.64×10^{-15}	84	3.10×10^{-16}
1×10^{-4}	1862	1.12×10^{-14}	81	5.49×10^{-16}
1×10^{-5}	2826	4.08×10^{-14}	81	3.98×10^{-16}
1×10^{-7}	11986	9.02×10^{-13}	81	3.98×10^{-16}

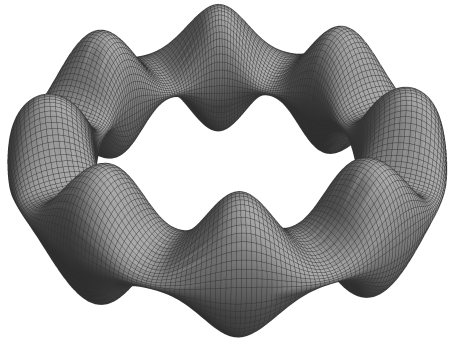
Laplace's equation on tori

$$\Delta u = 0 \quad \text{in } \Omega^c$$

$$\frac{\partial u}{\partial \nu} = g \quad \text{on } \partial\Omega$$

α	N_{tri}	N	N_{self}	T_{self}	T_{near}	T_{mult}	T_{total}	E
1.00	4	612	983	1.43×10^{-01}	$1.11 \times 10^{+00}$	2.18×10^{-01}	$1.60 \times 10^{+00}$	1.36×10^{-04}
	16	2448	978	5.83×10^{-01}	$2.84 \times 10^{+00}$	$3.44 \times 10^{+00}$	$7.86 \times 10^{+00}$	1.48×10^{-10}
	64	9792	976	$2.57 \times 10^{+00}$	$8.64 \times 10^{+00}$	$4.01 \times 10^{+01}$	$6.31 \times 10^{+01}$	8.64×10^{-14}
0.25	16	2448	976	5.60×10^{-01}	$1.63 \times 10^{+00}$	$3.23 \times 10^{+00}$	$6.31 \times 10^{+00}$	7.07×10^{-07}
	64	9792	976	$2.26 \times 10^{+00}$	$6.41 \times 10^{+00}$	$2.77 \times 10^{+01}$	$4.45 \times 10^{+01}$	2.19×10^{-11}
	256	39168	976	$8.88 \times 10^{+00}$	$2.54 \times 10^{+01}$	$1.38 \times 10^{+02}$	$2.19 \times 10^{+02}$	2.85×10^{-14}
0.10	40	6120	976	$1.42 \times 10^{+00}$	$3.34 \times 10^{+00}$	$1.09 \times 10^{+01}$	$1.92 \times 10^{+01}$	5.88×10^{-07}
	160	24480	975	$5.57 \times 10^{+00}$	$1.45 \times 10^{+01}$	$7.70 \times 10^{+01}$	$1.16 \times 10^{+02}$	1.11×10^{-11}
	640	97920	976	$2.23 \times 10^{+01}$	$5.91 \times 10^{+01}$	$5.00 \times 10^{+02}$	$6.92 \times 10^{+02}$	2.30×10^{-14}
0.01	400	61200	975	$1.38 \times 10^{+01}$	$2.86 \times 10^{+01}$	$1.05 \times 10^{+02}$	$1.79 \times 10^{+02}$	5.12×10^{-07}
	1600	244800	975	$5.54 \times 10^{+01}$	$1.32 \times 10^{+02}$	$8.47 \times 10^{+02}$	$1.20 \times 10^{+03}$	8.39×10^{-12}
	6400	979200	975	$2.22 \times 10^{+02}$	$5.94 \times 10^{+02}$	$4.76 \times 10^{+03}$	$6.70 \times 10^{+03}$	7.07×10^{-13}

Convergence study on a deformed torus



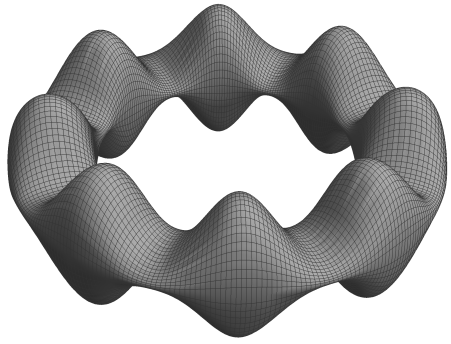
$$\Delta u + k^2 u = 0 \quad \text{in } \Omega$$
$$u = g \quad \text{on } \partial\Omega$$

Approximately 3 wavelengths in diameter.

N	Relative L^2 error potential	Relative L^2 error in normal derivative
576	0.289×10^{-01}	0.206×10^{-02}
2304	0.296×10^{-03}	0.191×10^{-02}
9216	0.267×10^{-05}	0.584×10^{-04}
36864	0.418×10^{-08}	0.752×10^{-06}

4^{th} order quadratures

Convergence study on a deformed torus



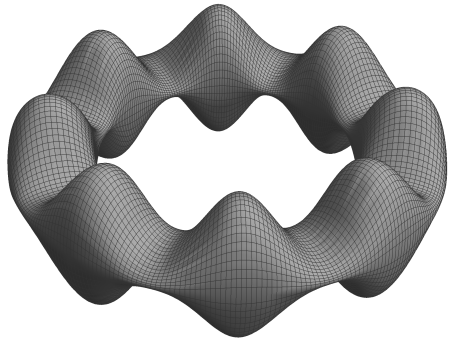
$$\Delta u + k^2 u = 0 \quad \text{in } \Omega$$
$$u = g \quad \text{on } \partial\Omega$$

Approximately 3 wavelengths in diameter.

N	Relative L^2 error potential	Relative L^2 error in normal derivative
1664	0.829×10^{-03}	0.303×10^{-03}
6656	0.159×10^{-05}	0.173×10^{-05}
26624	0.187×10^{-09}	0.159×10^{-07}
106496	0.480×10^{-13}	0.122×10^{-09}

8^{th} order quadratures

Convergence study on a deformed torus



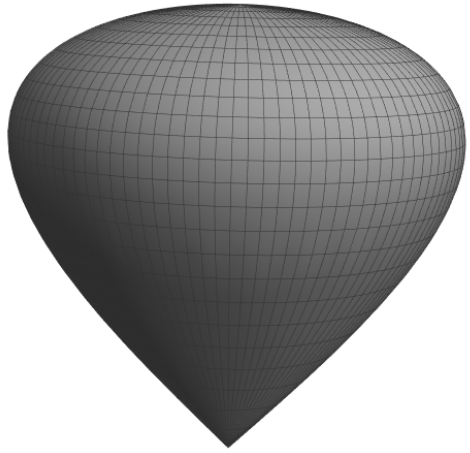
$$\Delta u + k^2 u = 0 \quad \text{in } \Omega$$
$$u = g \quad \text{on } \partial\Omega$$

Approximately 3 wavelengths in diameter.

N	Relative L^2 error potential	Relative L^2 error in normal derivative
3584	0.800×10^{-04}	0.103×10^{-03}
14336	0.155×10^{-07}	0.484×10^{-07}
57344	0.102×10^{-11}	0.213×10^{-08}
229376	0.957×10^{-13}	0.354×10^{-07}

12^{th} order quadratures

Sound-hard scattering from a snowcone



Approximately 3 wavelengths in diameter.

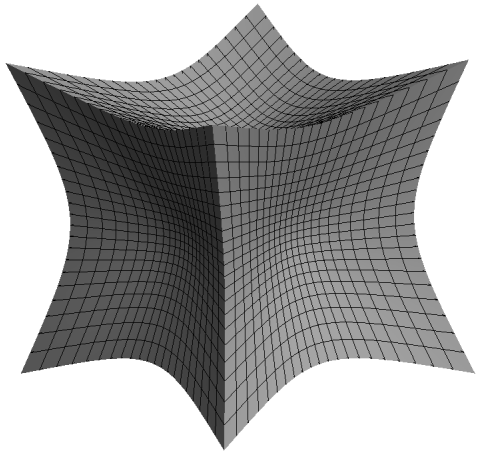
$$\Delta u + k^2 u = 0 \quad \text{in } \Omega^c$$

$$\frac{\partial u}{\partial \eta} = g \quad \text{on } \partial\Omega$$

$$|x| \left(\frac{\partial}{\partial |x|} - ik \right) u(x) \to 0 \text{ as } |x| \to \infty$$

N	κ	E
180	$2.40 \times 10^{+0}$	1.52×10^{-03}
720	$2.42 \times 10^{+0}$	2.42×10^{-05}
2880	$2.44 \times 10^{+0}$	1.04×10^{-07}
11520	$2.45 \times 10^{+0}$	9.09×10^{-10}
46080	$2.45 \times 10^{+0}$	7.04×10^{-13}

Sound-soft scattering from a deformed cube



$$\Delta u + k^2 u = 0 \quad \text{in } \Omega$$
$$u = g \quad \text{on } \partial\Omega$$

Approximately 8 wavelengths in diameter.

N_{tris}	N	Time	Error
192	32256	$1.23 \times 10^{+02}$	1.23×10^{-08}
432	72576	$4.77 \times 10^{+03}$	3.13×10^{-10}
768	129024	$6.21 \times 10^{+03}$	4.13×10^{-12}
1024	172032	$1.22 \times 10^{+04}$	7.13×10^{-14}

What's the deal with integral equation solvers?

- Integral equation methods are a mathematically elegant way to solve a fundamental problem regarding linear elliptic boundary value problems.
- Finally, an "off the shelf" integral equation solver which is competitive/superior to other methods for a certain class of problems has become available.
- That class of problems has some important applications, but is relatively narrow.
- A tremedous amount of work went into developing this solver and (most likely) a great deal more will be required to develop integral equation solvers for more general classes of problems.