ADAPTIVE MESHES AND EMBEDDED BOUNDARY INTEGRAL METHODS

Travis Askham (University of Washington)
March 15, 2018. ICERM workshop on “Fast Algorithms for Static and Dynamically Changing Point Configurations”
EMBEDDED BOUNDARY INTEGRAL METHODS

Collaborators:

Leslie Greengard

Manas Rachh

Ludvig af Klinteberg

Antoine Cerfon

Mary Catherine Kropinski

Bryan Quaife

INTEGRAL EQUATION METHODS FOR FLUIDS

Why integral equation methods?
- Geometric flexibility
- Well-conditioned formulations
- Existence of fast algorithms (FMM)

[Malhotra et al., 2017]

[Hoskins, Rachh, Serkh]

[Ojala, 2012]
Navier-Stokes

\[
\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \Delta \mathbf{u}, \quad \mathbf{x} \in \Omega
\]

\[
\nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in \Omega,
\]

\[
\mathbf{u} = \mathbf{f}, \quad \mathbf{x} \in \partial \Omega.
\]
NAVIER-STOKES TO MODIFIED STOKES

Navier-Stokes

\[
\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \Delta \mathbf{u}, \quad x \in \Omega
\]
\[
\nabla \cdot \mathbf{u} = 0, \quad x \in \Omega,
\]
\[
\mathbf{u} = \mathbf{f}, \quad x \in \partial \Omega.
\]

IMEX (Euler) Discretization

\[
\frac{\mathbf{u}^{N+1} - \mathbf{u}^N}{\delta t} - \frac{1}{\text{Re}} \Delta \mathbf{u}^{N+1} + \nabla p^{N+1} = \mathbf{F}, \quad x \in \Omega,
\]
\[
\nabla \cdot \mathbf{u}^{N+1} = 0, \quad x \in \Omega,
\]
\[
\mathbf{u}^{N+1} = \mathbf{f}, \quad x \in \partial \Omega.
\]
Let \( u^{N+1} = v + u_H \).

**Particular Solution \((v)\)**

\[
\begin{align*}
v - \frac{\delta t}{\text{Re}} \Delta v + \delta t \nabla p_v &= \delta t F + u^N, & \quad x \in \Omega, \\
\nabla \cdot v &= 0, & \quad x \in \Omega.
\end{align*}
\]
Let $u^{N+1} = v + u_H$.

**Particular Solution ($v$)**

$$ v - \frac{\delta t}{\text{Re}} \Delta v + \delta t \nabla p_V = \delta t F + u^N, \quad x \in \Omega, $$

$$ \nabla \cdot v = 0, \quad x \in \Omega. $$

**Boundary Correction ($u_H$) — Modified Stokes Equation**

$$ u_H - \frac{\delta t}{\text{Re}} \Delta u_H + \nabla p_H = 0, \quad x \in \Omega, $$

$$ \nabla \cdot u_H = 0, \quad x \in \Omega, $$

$$ u_H = f - v, \quad x \in \partial \Omega. $$
Let \( \lambda = \sqrt{\text{Re}/\delta t} \). The fundamental solution of the modified Stokes equations is the

**Modified Stokeslet**

\[
G(x, y) = (-\nabla^\perp \otimes \nabla^\perp) G(x, y),
\]

where

**Modified Biharmonic Green’s Function**

\[
G(x, y) = -\frac{1}{2\pi \lambda^2} \left( \log ||x - y|| + K_0(\lambda ||x - y||) \right).
\]
Let $\lambda = \sqrt{\text{Re}/\delta t}$. The fundamental solution of the modified Stokes equations is the

<table>
<thead>
<tr>
<th><strong>Modified Stokeslet</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$G(x, y) = (-\nabla_\perp \otimes \nabla_\perp )G(x, y)$,</td>
</tr>
</tbody>
</table>

where

<table>
<thead>
<tr>
<th><strong>Modified Biharmonic Green’s Function</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$G(x, y) = -\frac{1}{2\pi \lambda^2} (\log |x - y| + K_0(\lambda|x - y|))$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Particular Solution</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(x) = \int_\Omega G(x, y)(\delta t F(y) + u^N(y)) , dV(y)$</td>
</tr>
</tbody>
</table>

is a particular solution.
We represent the boundary correction \( u_H \) as a Double Layer Potential

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

where

\[
    D(x, y) = \nabla G_L(x, y) \otimes \nu + \nabla \perp \otimes \nabla \perp (\partial_\nu G(x, y)) + \nabla \perp \otimes \nabla (\partial_\tau G(x, y)).
\]

Get a second kind integral equation (SKIE) for \( \sigma \). This is a good thing!
EVALUATING THE BOUNDARY CORRECTION
For good performance, need:

**Figure:** Visualization of QBX idea. Taken from Klöckner, et al. 2012.
For good performance, need:

- High-order accurate quadrature for singular integrals (e.g. generalized Gaussian quadrature)

Figure: Visualization of QBX idea. Taken from Klöckner, et al. 2012.
For good performance, need:

- High-order accurate quadrature for singular integrals (e.g. generalized Gaussian quadrature)
- Fast solution methods for structured, dense linear systems (e.g. HSS, HODLR, GMRES)

**Figure:** Visualization of QBX idea. Taken from Klöckner, et al. 2012.
For good performance, need:

- High-order accurate quadrature for singular integrals (e.g. generalized Gaussian quadrature)
- Fast solution methods for structured, dense linear systems (e.g. HSS, HODLR, GMRES)
- Fast, accurate layer potential evaluation, including near-singular points (e.g. quadrature by expansion)

*Figure:* Visualization of QBX idea. Taken from Klöckner, et al. 2012.
To implement an integral equation method (both fast solvers and fast QBX), we need to be able to compute sums of the form

\[ u(x_i) = \sum_{j=1}^{n} q_j \partial_{v_j w_j} G(x_i, s_j) \]

quickly and stably (and its derivatives)
To implement an integral equation method (both fast solvers and fast QBX), we need to be able to compute sums of the form

\[ u(x_i) = \sum_{j=1}^{n} q_j \partial_{v_jw_j} g(x_i, s_j) \]

generally quickly and stably (and its derivatives).

Let

\[ A = \begin{pmatrix}
\partial_{v_1w_1} g(x_1, s_1) & \partial_{v_2w_2} g(x_1, s_2) & \cdots & \partial_{v_nw_n} g(x_1, s_n) \\
\partial_{v_1w_1} g(x_2, s_1) & \partial_{v_2w_2} g(x_2, s_2) & \cdots & \partial_{v_nw_n} g(x_2, s_n) \\
\vdots & \vdots & \ddots & \vdots \\
\partial_{v_1w_1} g(x_m, s_1) & \partial_{v_2w_2} g(x_m, s_2) & \cdots & \partial_{v_nw_n} g(x_m, s_n)
\end{pmatrix} \]
LOW-RANK INTERACTIONS

\[
A = \begin{pmatrix}
\partial_{v_1 w_1} G(x_1, s_1) & \partial_{v_2 w_2} G(x_1, s_2) & \cdots & \partial_{v_n w_n} G(x_1, s_n) \\
\partial_{v_1 w_1} G(x_2, s_1) & \partial_{v_2 w_2} G(x_2, s_2) & \cdots & \partial_{v_n w_n} G(x_2, s_n) \\
\vdots & \vdots & \ddots & \vdots \\
\partial_{v_1 w_1} G(x_m, s_1) & \partial_{v_2 w_2} G(x_m, s_2) & \cdots & \partial_{v_n w_n} G(x_m, s_n)
\end{pmatrix}
\]

Well-separated points
LOW-RANK INTERACTIONS

\[ A = \begin{pmatrix}
\frac{\partial}{\partial v_1 w_1} g(x_1, s_1) & \frac{\partial}{\partial v_2 w_2} g(x_1, s_2) & \cdots & \frac{\partial}{\partial v_n w_n} g(x_1, s_n) \\
\frac{\partial}{\partial v_1 w_1} g(x_2, s_1) & \frac{\partial}{\partial v_2 w_2} g(x_2, s_2) & \cdots & \frac{\partial}{\partial v_n w_n} g(x_2, s_n) \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial}{\partial v_1 w_1} g(x_m, s_1) & \frac{\partial}{\partial v_2 w_2} g(x_m, s_2) & \cdots & \frac{\partial}{\partial v_n w_n} g(x_m, s_n)
\end{pmatrix} \]

Well-separated points

The rank is low, independent of number of sources and targets.

For certain kernels, low-rank decompositions are known analytically.

singualar values of $A$ for various values of $m$ and $n$
LOW-RANK INTERACTIONS

\[ A = \begin{pmatrix} \partial v_1 w_1 G(x_1, s_1) & \partial v_2 w_2 G(x_1, s_2) & \cdots & \partial v_n w_n G(x_1, s_n) \\ \partial v_1 w_1 G(x_2, s_1) & \partial v_2 w_2 G(x_2, s_2) & \cdots & \partial v_n w_n G(x_2, s_n) \\ \vdots & \vdots & \ddots & \vdots \\ \partial v_1 w_1 G(x_m, s_1) & \partial v_2 w_2 G(x_m, s_2) & \cdots & \partial v_n w_n G(x_m, s_n) \end{pmatrix} \]

Well-separated points

- The rank is low, independent of number of sources and targets

![Graph showing singular values of A for various values of m and n](image)

singular values of A for various values of m and n
LOW-RANK INTERACTIONS

\[ A = \begin{pmatrix}
\frac{\partial v_1 w_1}{\partial v_1} G(x_1, s_1) & \frac{\partial v_1 w_2}{\partial v_2} G(x_1, s_2) & \cdots & \frac{\partial v_1 w_n}{\partial v_n} G(x_1, s_n) \\
\frac{\partial v_2 w_1}{\partial v_2} G(x_2, s_1) & \frac{\partial v_2 w_2}{\partial v_2} G(x_2, s_2) & \cdots & \frac{\partial v_2 w_n}{\partial v_n} G(x_2, s_n) \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial v_m w_1}{\partial v_m} G(x_m, s_1) & \frac{\partial v_m w_2}{\partial v_m} G(x_m, s_2) & \cdots & \frac{\partial v_m w_n}{\partial v_n} G(x_m, s_n)
\end{pmatrix} \]

Well-separated points

- The rank is low, independent of number of sources and targets
- For certain kernels, low-rank decompositions are known analytically

Graph of singular values of \( A \) for various values of \( m \) and \( n \)
NUMERICAL INSTABILITY

\[ G(x, y) = -\frac{1}{2\pi \lambda^2} (\log \|x - y\| + K_0(\lambda \|x - y\|)) . \]

Why not use existing tech for \( \log \) and \( K_0 \) and add together?
NUMERICAL INSTABILITY

\[ G(x, y) = -\frac{1}{2\pi \lambda^2} \left( \log \|x - y\| + K_0(\lambda\|x - y\|) \right). \]

Why not use existing tech for log and \( K_0 \) and add together?

**Numerical Experiment**

\[ u(x; \lambda) = \sum_{j=1}^{n_s} q_j \partial_{v_j w_j} G(x, s_j), \]
NUMERICAL INSTABILITY

\[ G(x, y) = -\frac{1}{2\pi \lambda^2} \left( \log \|x - y\| + K_0(\lambda \|x - y\|) \right) . \]

Why not use existing tech for \( \log \) and \( K_0 \) and add together?

**Numerical Experiment**

\[ u(x; \lambda) = \sum_{j=1}^{n_s} q_j \partial_{v_j w_j} G(x, s_j) , \]

\[ u_L(x; \lambda) = -\frac{1}{2\pi \lambda^2} \sum_{j=1}^{n_s} q_j \partial_{v_j w_j} \log \|x - s_j\| , \]
NUMERICAL INSTABILITY

\[ G(x, y) = -\frac{1}{2\pi \lambda^2} (\log ||x - y|| + K_0(\lambda ||x - y||)) . \]

Why not use existing tech for \( \log \) and \( K_0 \) and add together?

**Numerical Experiment**

\[ u(x; \lambda) = \sum_{j=1}^{n_s} q_j \partial_{v_j w_j} G(x, s_j) , \]

\[ u_L(x; \lambda) = -\frac{1}{2\pi \lambda^2} \sum_{j=1}^{n_s} q_j \partial_{v_j w_j} \log ||x - s_j|| , \]

\[ u_K(x; \lambda) = \frac{1}{2\pi \lambda^2} \sum_{j=1}^{n_s} q_j \partial_{v_j w_j} K_0(\lambda ||x - s_j||) . \]
\[ G(x, y) = -\frac{1}{2\pi \lambda^2} \left( \log \| x - y \| + K_0(\lambda \| x - y \|) \right). \]

Why not use existing tech for log and \( K_0 \) and add together?

**Numerical Experiment**

\[ u(x; \lambda) = \sum_{j=1}^{n_s} q_j \partial_{v_j w_j} G(x, s_j), \]

\[ u_L(x; \lambda) = -\frac{1}{2\pi \lambda^2} \sum_{j=1}^{n_s} q_j \partial_{v_j w_j} \log \| x - s_j \|, \]

\[ u_K(x; \lambda) = \frac{1}{2\pi \lambda^2} \sum_{j=1}^{n_s} q_j \partial_{v_j w_j} K_0(\lambda \| x - s_j \|). \]

What is the error (in floating point) in evaluating \( u \) as \( u = u_L - u_K \)?
NUMERICAL INSTABILITY (CONT.)

\[ G(x, y) = -\frac{1}{2\pi \lambda^2} \left( \log \|x - y\| + K_0(\lambda \|x - y\|) \right). \]

Why not use existing tech for log and \( K_0 \) and add together?

(a) Interior problem

(b) Exterior problem

The error increases as the product of \( \lambda = \sqrt{\text{Re}/\delta t} \) and the radius of the disc \( R \) goes to zero.
Note that $\lambda = \sqrt{\text{Re}/\delta t}$.
The meaning of $\lambda R$

Note that $\lambda = \sqrt{\text{Re}/\delta t}$

The value of $\lambda R$ is small if

- The Reynolds number is small (viscous fluids)
- The grid is fine
- Time steps are relatively long

Note that $\lambda R < 1$ when $\delta t > \text{Re}/R^2$, i.e. when the CFL condition is violated. This regime is important for implicit methods for viscous fluids.
Note that $\lambda = \sqrt{Re/\delta t}$

The value of $\lambda R$ is small if

- The Reynolds number is small (viscous fluids)
THE MEANING OF $\lambda R$

Note that $\lambda = \sqrt{\text{Re}/\delta t}$

The value of $\lambda R$ is small if

- The Reynolds number is small (viscous fluids)
- The grid is fine
THE MEANING OF $\lambda R$

Note that $\lambda = \sqrt{\text{Re}/\delta t}$

The value of $\lambda R$ is small if

- The Reynolds number is small (viscous fluids)
- The grid is fine
- Time steps are relatively long
THE MEANING OF $\lambda R$

Note that $\lambda = \sqrt{\frac{Re}{\delta t}}$

The value of $\lambda R$ is small if

- The Reynolds number is small (viscous fluids)
- The grid is fine
- Time steps are relatively long

Note that $\lambda R < 1$ when $\delta t > ReR^2$, i.e. when the CFL condition is violated. This regime is important for implicit methods for viscous fluids.
Our goal: analytical formulas for the low rank interaction between well separated points which are stable for any $\lambda R$. 
OUR GOAL

Our goal: analytical formulas for the low rank interaction between well separated points which are stable for any $\lambda R$.

Go back to basics: look that the separation of variables problem for the modified biharmonic equation
Let $\Omega$ be the interior or exterior of a disc of radius $R$ and consider the modified biharmonic equation:

$$\Delta(\Delta - \lambda^2)u = 0, \quad x \in \Omega,$$

$$u = f, \quad \partial_n u = g, \quad x \in \partial \Omega.$$
Let $\Omega$ be the interior or exterior of a disc of radius $R$ and consider the modified biharmonic equation:

$$\Delta (\Delta - \lambda^2) u = 0, \ x \in \Omega,$$

$$u = f, \partial_n u = g, \ x \in \partial \Omega.$$

**Separation of Variables Representation**

$$u(r, \theta) = \sum_{n=-\infty}^{\infty} u_n(r) e^{i n \theta}.$$
SEPARATION OF VARIABLES

Let $\Omega$ be the interior or exterior of a disc of radius $R$ and consider the modified biharmonic equation:

$$\Delta(\Delta - \lambda^2)u = 0, \quad x \in \Omega,$$

$$u = f, \quad \partial_n u = g, \quad x \in \partial \Omega.$$

**Separation of Variables Representation**

$$u(r, \theta) = \sum_{n=-\infty}^{\infty} u_n(r) e^{in\theta}.$$

**ODE for $u_n(r)$**

$$\left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{n^2}{r^2} \right) \left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{n^2}{r^2} - \lambda^2 \right) u_n(r) = 0.$$
ODE for \( u_n(r) \)

\[
\left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{n^2}{r^2} \right) \left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{n^2}{r^2} - \lambda^2 \right) u_n(r) = 0.
\]

Four linearly independent solutions: \( r^{|n|}, I_n(\lambda r), r^{-|n|}, \) and \( K_n(\lambda r). \)
SEPARATION OF VARIABLES (CONT.)

ODE for $u_n(r)$

$$\left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{n^2}{r^2} \right) \left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{n^2}{r^2} - \lambda^2 \right) u_n(r) = 0.$$ 

Four linearly independent solutions: $r^{\pm n}$, $I_n(\lambda r)$, and $K_n(\lambda r)$.

Interior Problem

By imposing continuity at $r = 0$, the functions $r^{\pm n}$ and $I_n(\lambda r)$ are a basis for the interior problem.
ODE for $u_n(r)$

\[
\left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{n^2}{r^2} \right) \left( \frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} - \frac{n^2}{r^2} - \lambda^2 \right) u_n(r) = 0 .
\]

Four linearly independent solutions: $r^{|n|}$, $I_n(\lambda r)$, $r^{-|n|}$, and $K_n(\lambda r)$.

**Interior Problem**

By imposing continuity at $r = 0$, the functions $r^{|n|}$ and $I_n(\lambda r)$ are a basis for the interior problem.

**Exterior Problem**

By imposing decay conditions $r = \infty$, the functions $r^{-|n|}$ and $K_n(\lambda r)$ are a basis for the exterior problem.
For the exterior problem, we have $u_n(r) = \alpha_n r^{-|n|} + \beta_n K_n(\lambda r)$.
For the exterior problem, we have \( u_n(r) = \alpha_n r^{-|n|} + \beta_n K_n(\lambda r) \).

**Coefficient Recovery Problem**

\[
\begin{pmatrix}
R^{-|n|} & K_n(\lambda R) \\
-|n| R^{-|n|-1} & -\frac{\lambda}{2} (K_{n-1}(\lambda R) + K_{n+1}(\lambda R))
\end{pmatrix}
\begin{pmatrix}
\alpha_n \\
\beta_n
\end{pmatrix}
= 
\begin{pmatrix}
f_n \\
g_n
\end{pmatrix}.
\]

This problem is ill-conditioned for small \( \lambda R \). Intuitively, this is because \( K_n(\lambda r) \) and \( r^{-|n|} \) are similar functions for small \( r \).
For the exterior problem, we have $u_n(r) = \alpha_n r^{-|n|} + \beta_n K_n(\lambda r)$.

**Coefficient Recovery Problem**

\[
\begin{pmatrix}
R^{-|n|} & K_n(\lambda R) \\
-|n| R^{-|n|-1} & -\frac{\lambda}{2} (K_{n-1}(\lambda R) + K_{n+1}(\lambda R))
\end{pmatrix}
\begin{pmatrix}
\alpha_n \\
\beta_n
\end{pmatrix}
= \begin{pmatrix}
f_n \\
g_n
\end{pmatrix}.
\]

This problem is ill-conditioned for small $\lambda R$. Intuitively, this is because $K_n(\lambda r)$ and $r^{-|n|}$ are similar functions for small $r$.

**Asymptotic Expansion for $K_n(\lambda r)$**

\[
K_n(\lambda r) = \frac{1}{2}(\frac{1}{2} \lambda r)^{-|n|} \sum_{k=0}^{|n|-1} \frac{|n| - k - 1)!}{k!} (\frac{1}{4} \lambda r^2)^k + (-1)^{|n|+1} \ln (\frac{1}{2} \lambda r) I_n (\lambda r)
\]

\[+ (-1)^{|n|} \frac{1}{2}(\frac{1}{2} \lambda r)^{|n|} \sum_{k=0}^{\infty} \left( \psi (k + 1) + \psi (|n| + k + 1) \right) \frac{(\frac{1}{4} \lambda r^2)^k}{k!(|n| + k)!} \cdot
\]
We can define a new basis function for the exterior problem which is not asymptotically similar to $r^{-|n|}$ and $K_n$. 

\[ Q_n(r) = K_n(\lambda r) - 2|n|^{-1}(|n|^{-1})! \lambda |n| r |n|. \]

$Q_n$ has a different leading order term for small $\lambda$ and $R$. The pair $(Q_n, K_n)$ is a better conditioned basis than $(r^{-|n|}, K_n)$ in the small $\lambda$ regime. $Q_n$ is still a solution of the ODE for $u_n$ because it's a linear combo of $r^{-|n|}$ and $K_n$. It is simple to evaluate $Q_n$ with tweaks to existing software.
We can define a new basis function for the exterior problem which is *not* asymptotically similar to $r^{-|n|}$ and $K_n$.

**Definition of $Q_n$**

$$Q_n(r) = K_n(\lambda r) - \frac{2^{|n|-1}(|n| - 1)!}{\lambda |n| r^n}.$$
We can define a new basis function for the exterior problem which is not asymptotically similar to \( r^{-|n|} \) and \( K_n \).

**Definition of \( Q_n \)**

\[
Q_n(r) = K_n(\lambda r) - \frac{2^{|n|-1}(|n|-1)!}{\lambda^{|n|}r^{|n|}}.
\]

- \( Q_n \) has a different leading order term for small \( \lambda \) and \( R \).
We can define a new basis function for the exterior problem which is *not* asymptotically similar to $r^{-|n|}$ and $K_n$.

**Definition of $Q_n$**

$$Q_n(r) = K_n(\lambda r) - \frac{2^{|n|-1}(|n| - 1)!}{\lambda |n| r |n|}. $$

- $Q_n$ has a different leading order term for small $\lambda$ and $R$.
- The pair $(Q_n, K_n)$ is a better conditioned basis than $(r^{-|n|}, K_n)$ in the small $\lambda R$ regime.
We can define a new basis function for the exterior problem which is \textit{not} asymptotically similar to \( r^{-|n|} \) and \( K_n \).

**Definition of \( Q_n \)**

\[
Q_n(r) = K_n(\lambda r) - \frac{2^{|n|-1} (|n| - 1)!}{\lambda^{|n|} r^{|n|}}.
\]

- \( Q_n \) has a different leading order term for small \( \lambda \) and \( R \).
- The pair \((Q_n, K_n)\) is a better conditioned basis than \((r^{-|n|}, K_n)\) in the small \( \lambda R \) regime.
- \( Q_n \) is still a solution of the ODE for \( u_n \) because it’s a linear combo of \( r^{-|n|} \) and \( K_n \).
We can define a new basis function for the exterior problem which is not asymptotically similar to $r^{-|n|}$ and $K_n$.

**Definition of $Q_n$**

$$Q_n(r) = K_n(\lambda r) - \frac{2^{|n|-1} (|n| - 1)!}{\lambda^{|n|} r^{|n|}}.$$

- $Q_n$ has a different leading order term for small $\lambda$ and $R$.
- The pair $(Q_n, K_n)$ is a better conditioned basis than $(r^{-|n|}, K_n)$ in the small $\lambda R$ regime.
- $Q_n$ is still a solution of the ODE for $u_n$ because it’s a linear combo of $r^{-|n|}$ and $K_n$.
- It is simple to evaluate $Q_n$ with tweaks to existing software.
For the interior problem, we have that $u_n(r) = \alpha_n r^{|\nu|} + \beta_n I_n(\lambda r)$.
For the interior problem, we have that $u_n(r) = \alpha_n r^{|n|} + \beta_n I_n(\lambda r)$.

**Coefficient Recovery Problem**

$$
\begin{pmatrix}
R^{|n|} \\
|n|R^{|n|}^{-1} \\
\frac{\lambda}{2} (I_{n-1}(\lambda R) + I_{n+1}(\lambda R))
\end{pmatrix} \begin{pmatrix}
\alpha_n \\
\beta_n
\end{pmatrix} = \begin{pmatrix}
f_n \\
g_n
\end{pmatrix}.
$$

This problem is again ill-conditioned for small $\lambda R$. 

For the interior problem, we have that $u_n(r) = \alpha_n r^{|n|} + \beta_n l_n(\lambda r)$.

**Coefficient Recovery Problem**

$$
\begin{pmatrix}
R^{|n|} & l_n(\lambda R) \\
|n|R^{|n|}-1 & \frac{\lambda}{2} (l_{n-1}(\lambda R) + l_{n+1}(\lambda R))
\end{pmatrix}
\begin{pmatrix}
\alpha_n \\
\beta_n
\end{pmatrix}
= 
\begin{pmatrix}
 f_n \\
g_n
\end{pmatrix}.
$$

This problem is again ill-conditioned for small $\lambda R$.

**Asymptotic Expansion for $l_n(\lambda r)$**

$$
l_n(\lambda r) = \sum_{k=0}^{\infty} \frac{\left(\frac{\lambda r}{2}\right)^{2k+|n|}}{k!(k+|n|)!} = \frac{1}{2^{|n|}|n|!} (\lambda r)^{|n|} + \frac{1}{2^{|n|+2}(|n|+1)!} (\lambda r)^{|n|+2} + \cdots
$$
Again, we can define a new basis function for the interior problem which is \textit{not} asymptotically similar to \( r^n \) and \( I_n \).
Again, we can define a new basis function for the interior problem which is \textit{not} asymptotically similar to $r|n|$ and $I_n$.

\textbf{Definition of $P_n$}

\begin{equation}
P_n(r) = I_n(\lambda r) - \left(\frac{\lambda r}{2}\right)^{|n|} \frac{1}{|n|!}.
\end{equation}
A BETTER BASIS (INT.)

Again, we can define a new basis function for the interior problem which is *not* asymptotically similar to $r^{n}$ and $I_{n}$.

**Definition of $P_{n}$**

\[ P_{n}(r) = I_{n}(\lambda r) - \left( \frac{\lambda r}{2} \right)^{|n|} \frac{1}{|n|!}. \]

- $P_{n}$ has a different leading order term for small $\lambda$ and $R$. 
Again, we can define a new basis function for the interior problem which is \textit{not} asymptotically similar to $r^{|n|}$ and $I_n$.

\textbf{Definition of $P_n$}

\[ P_n(r) = I_n(\lambda r) - \left( \frac{\lambda r}{2} \right)^{|n|} \frac{1}{|n|!}. \]

- $P_n$ has a different leading order term for small $\lambda$ and $R$.
- The pair $(r^{|n|}, P_n)$ is a better conditioned basis than $(r^{|n|}, I_n)$ in the small $\lambda R$ regime.
Again, we can define a new basis function for the interior problem which is \textit{not} asymptotically similar to \( r^n \) and \( I_n \).

**Definition of \( P_n \)**

\[
P_n(r) = I_n(\lambda r) - \left( \frac{\lambda r}{2} \right)^{|n|} \frac{1}{|n|!}.
\]

- \( P_n \) has a different leading order term for small \( \lambda \) and \( R \).
- The pair \((r^n, P_n)\) is a better conditioned basis than \((r^n, I_n)\) in the small \( \lambda R \) regime.
- \( P_n \) is still a solution of the ODE for \( u_n \) because it’s a linear combo of \( r^n \) and \( I_n \).
Again, we can define a new basis function for the interior problem which is *not* asymptotically similar to $r^n$ and $I_n$.

**Definition of $P_n$**

$$P_n(r) = I_n(\lambda r) - \left( \frac{\lambda r}{2} \right)^{|n|} \frac{1}{|n|!}.$$ 

- $P_n$ has a different leading order term for small $\lambda$ and $R$.
- The pair $(r^n, P_n)$ is a better conditioned basis than $(r^n, I_n)$ in the small $\lambda R$ regime.
- $P_n$ is still a solution of the ODE for $u_n$ because it’s a linear combo of $r^n$ and $I_n$.
- It is simple to evaluate $P_n$ with tweaks to existing software.
Question
What is the practical effect of the condition number of the coefficient recovery problem on the accuracy of the solution?
Numerical Experiment

\[ u(x; \lambda) = \sum_{j=1}^{n_s} \lambda^2 c_j \mathcal{G}(x, s_j) + \lambda d_j \partial_{\nu_j,1} \mathcal{G}(x, s_j) \]
\[ + q_j \partial_{\nu_j,2} \nu_j,3 \mathcal{G}(x, s_j). \]

For several values of \( \lambda \) and \( R \):

Evaluate \( u \) and \( \partial u \) on \( \partial \Omega \)

Solve corresponding separation of variables problem (order \( N = 50 \), using 100 points on \( \partial \Omega \)) with new and old basis

Evaluate error in potential, gradient, and Hessian

Should be good to about machine precision, with some precision loss in the derivatives
**Numerical Experiment**

\[ u(x; \lambda) = \sum_{j=1}^{n_s} \lambda^2 c_j G(x, s_j) + \lambda d_j \partial_{y_1} G(x, s_j) + q_j \partial_{y_2} G(x, s_j). \]

For several values of \( \lambda \) and \( R \):
- Evaluate \( u \) and \( \partial_n u \) on \( \partial \Omega \)
Numerical Experiment

\[
 u(x; \lambda) = \sum_{j=1}^{n_s} \lambda^2 c_j G(x, s_j) + \lambda d_j \partial_{v_j,1} G(x, s_j) \\
+ q_j \partial_{v_j,2} v_j,3 G(x, s_j).
\]

For several values of \( \lambda \) and \( R \):

- Evaluate \( u \) and \( \partial_n u \) on \( \partial \Omega \)
- Solve corresponding separation of variables problem (order \( N = 50 \), using 100 points on \( \partial \Omega \)) with new and old basis functions
Numerical Experiment

\[ u(x; \lambda) = \sum_{j=1}^{n_s} \lambda^2 c_j G(x, s_j) + \lambda d_j \partial_{\nu_j,1} G(x, s_j) + q_j \partial_{\nu_j,2} v_j,3 G(x, s_j). \]

For several values of \( \lambda \) and \( R \):

- Evaluate \( u \) and \( \partial_n u \) on \( \partial \Omega \)
- Solve corresponding separation of variables problem (order \( N = 50 \), using 100 points on \( \partial \Omega \)) with new and old basis functions
- Evaluate error in potential, gradient, and Hessian
NUMERICAL RESULTS (CONT.)

**Numerical Experiment**

\[ u(x; \lambda) = \sum_{j=1}^{n_s} \lambda^2 c_j g(x, s_j) + \lambda d_j \partial_{v_j,1} g(x, s_j) \]

\[ + q_j \partial_{v_j,2} v_j,3 g(x, s_j). \]

For several values of \( \lambda \) and \( R \):

- Evaluate \( u \) and \( \partial_n u \) on \( \partial \Omega \)
- Solve corresponding separation of variables problem (order \( N = 50 \), using 100 points on \( \partial \Omega \)) with new and old basis functions
- Evaluate error in potential, gradient, and Hessian
- Should be good to about machine precision, with some precision loss in the derivatives
Errors for the exterior problem: $(r^{-|n|}, K_n)$ vs $(Q_n, K_n)$. Top row: $\lambda \to 0$. Bottom row: $R \to 0$. 
Errors for the interior problem: \((r|n|, I_n)\) vs \((r|n|, P_n)\). Top row: \(\lambda \rightarrow 0\). Bottom row: \(R \rightarrow 0\).
REALITY CHECK

How is this a decomposition?
REALITY CHECK

How is this a decomposition? Recall

\[ u(x_i) = \sum_{j=1}^{n} q_j \partial_{v_j w_j} G(x_i, s_j) \]

\[ A = \begin{pmatrix}
\partial_{v_1 w_1} G(x_1, s_1) & \partial_{v_2 w_2} G(x_1, s_2) & \cdots & \partial_{v_n w_n} G(x_1, s_n) \\
\partial_{v_1 w_1} G(x_2, s_1) & \partial_{v_2 w_2} G(x_2, s_2) & \cdots & \partial_{v_n w_n} G(x_2, s_n) \\
\vdots & \vdots & \ddots & \vdots \\
\partial_{v_1 w_1} G(x_m, s_1) & \partial_{v_2 w_2} G(x_m, s_2) & \cdots & \partial_{v_n w_n} G(x_m, s_n)
\end{pmatrix} \]

Well-separated points
$A = LR^T$. 
$$A = L R^T.$$ The form of $L$ is straightforward

$$L = \begin{pmatrix} Q_0(|x_1 - c|) & K_0(\lambda|x_1 - c|) & \cdots & Q_p(|x_1 - c|) e^{i p \theta_1} & K_p(\lambda|x_1 - c|) e^{i p \theta_1} \\ Q_0(|x_2 - c|) & K_0(\lambda|x_2 - c|) & \cdots & Q_p(|x_2 - c|) e^{i p \theta_2} & K_p(\lambda|x_2 - c|) e^{i p \theta_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ Q_0(|x_m - c|) & K_0(\lambda|x_m - c|) & \cdots & Q_p(|x_m - c|) e^{i p \theta_m} & K_p(\lambda|x_m - c|) e^{i p \theta_m} \end{pmatrix}$$

What is $R^T$? It is the map from the sources to the coefficients. $R^T$ is solved for $2 \times 2$ for coeffs. Separate modes with FFT. Evaluate both $u$ and $\partial_n u$ on disc boundary.

Note that there is an analytical formula for $R^T$ [Askham, 2017].
$A = LR^\top$. The form of $L$ is straightforward

$$L = \begin{pmatrix}
Q_0(|x_1 - c|) & K_0(\lambda|x_1 - c|) & \cdots & Q_p(|x_1 - c|)e^{ip\theta_1} & K_p(\lambda|x_1 - c|)e^{ip\theta_1} \\
Q_0(|x_2 - c|) & K_0(\lambda|x_2 - c|) & \cdots & Q_p(|x_2 - c|)e^{ip\theta_2} & K_p(\lambda|x_2 - c|)e^{ip\theta_2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
Q_0(|x_m - c|) & K_0(\lambda|x_m - c|) & \cdots & Q_p(|x_m - c|)e^{ip\theta_m} & K_p(\lambda|x_m - c|)e^{ip\theta_m}
\end{pmatrix}$$

What is $R^\top$?
ANALYTICAL DECOMPOSITION

\[ A = LR^T. \] The form of \( L \) is straightforward

\[
L = \begin{pmatrix}
Q_0(|x_1 - c|) & K_0(\lambda|x_1 - c|) & \cdots & Q_p(|x_1 - c|)e^{ip\theta_1} & K_p(\lambda|x_1 - c|)e^{ip\theta_1} \\
Q_0(|x_2 - c|) & K_0(\lambda|x_2 - c|) & \cdots & Q_p(|x_2 - c|)e^{ip\theta_2} & K_p(\lambda|x_2 - c|)e^{ip\theta_2} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
Q_0(|x_m - c|) & K_0(\lambda|x_m - c|) & \cdots & Q_p(|x_m - c|)e^{ip\theta_m} & K_p(\lambda|x_m - c|)e^{ip\theta_m}
\end{pmatrix}
\]

What is \( R^T \)? It is the map from the sources to the coefficients

\[
R^T = \begin{array}{|c|c|c|}
\text{each mode} & \text{separate modes with FFT} & \text{evaluate both } u \text{ and } \partial_n u \text{ on disc boundary} \\
\text{solve } 2 \times 2 \text{ for coeffs} & & \\
\end{array}
\]

Note that there is an analytical formula for \( R^T \) [Askham, 2017].
Because the formulas for $L$ and $R^\top$ are known, forming these matrices is $O((m + n)p)$. 
WHAT OF EFFICIENCY?

Because the formulas for $L$ and $R^T$ are known, forming these matrices is $O((m + n)p)$. The SVD, on the other hand is $O(mn^2)$. 

WHAT OF EFFICIENCY?

Because the formulas for $L$ and $R^T$ are known, forming these matrices is $O((m + n)p)$. The SVD, on the other hand is $O(mn^2)$. Even randomized methods for the SVD are $O(mnp)$. 
Because the formulas for $L$ and $R^T$ are known, forming these matrices is $O((m + n)p)$. The SVD, on the other hand is $O(mn^2)$. Even randomized methods for the SVD are $O(mnp)$.

It is not always the case that sources are well-separated from targets. Can we make a stable FMM with the above?
The preceding provides a stable fast multipole method

A fast multipole method is based on:
The preceding provides a stable fast multipole method

A fast multipole method is based on:

1. a formula for representing the sum due to a localized subset of the points (a multipole expansion). \((Q_n, K_n)\)
The preceding provides a stable fast multipole method.

A fast multipole method is based on:

1. a formula for representing the sum due to a localized subset of the points (a multipole expansion). \((Q_n, K_n)\)

2. a formula for representing the sum due to points outside of a disc (a local expansion). \((r^{\mid n\mid}, P_n)\)
The preceding provides a stable fast multipole method.

A fast multipole method is based on:

1. a formula for representing the sum due to a localized subset of the points (a multipole expansion). \((Q_n, K_n)\)

2. a formula for representing the sum due to points outside of a disc (a local expansion). \((r^n, P_n)\)

3. formulas for translating between these representations (translation operators). see the preprint!

AN FMM
The preceding provides a stable fast multipole method

A fast multipole method is based on:

1. a formula for representing the sum due to a localized subset of the points (a multipole expansion). \((Q_n, K_n)\)
2. a formula for representing the sum due to points outside of a disc (a local expansion). \((r|n|, P_n)\)
3. formulas for translating between these representations (translation operators). see the preprint!
4. a hierarchical organization of source and target points in space
COMPUTING THE PARTICULAR SOLUTION
To compute the particular solution, we need to evaluate integrals of the form

\[ v(x) = Vf(x) := \int_{\Omega} K(x, y)f(y) \, dy, \]

where \( K(x, y) = \log |x - y| \) or \( K(x, y) = K_0(\lambda|x - y|) \).
To compute the particular solution, we need to evaluate integrals of the form

\[ v(x) = Vf(x) := \int_{\Omega} K(x, y)f(y) \, dy, \]

where \( K(x, y) = \log |x - y| \) or \( K(x, y) = K_0(\lambda |x - y|). \)

- No solve, just apply
To compute the particular solution, we need to evaluate integrals of the form

\[ v(x) = Vf(x) := \int_{\Omega} K(x, y)f(y) \, dy, \]

where \( K(x, y) = \log |x - y| \) or \( K(x, y) = K_0(\lambda |x - y|) \).

- No solve, just apply
- Weakly singular integrand
To compute the particular solution, we need to evaluate integrals of the form

$$v(x) = Vf(x) := \int_{\Omega} K(x, y)f(y) \, dy,$$

where $K(x, y) = \log |x - y|$ or $K(x, y) = K_0(\lambda|x - y|)$. 

- No solve, just apply
- Weakly singular integrand
- Expensive on an unstructured discretization (adaptive quadrature, etc.)
To compute the particular solution, we need to evaluate integrals of the form

\[ v(x) = Vf(x) := \int_{\Omega} \mathcal{K}(x, y)f(y) \, dy, \]

where \( \mathcal{K}(x, y) = \log |x - y| \) or \( \mathcal{K}(x, y) = K_0(\lambda |x - y|) \).

- No solve, just apply
- Weakly singular integrand
- Expensive on an unstructured discretization (adaptive quadrature, etc.)
- Fast methods for regular domains
  - Disc solvers
  - “Box codes” (Ethridge and Greengard, Cheng et al., Langston and Zorin)
Box codes (typically) work on level-restricted trees and are very efficient (density $f$ defined on leaves):
Box codes (typically) work on level-restricted trees and are very efficient (density $f$ defined on leaves):

- Limited number of possible local interactions (precomputation of integrals to near machine precision)
Box codes (typically) work on level-restricted trees and are very efficient (density $f$ defined on leaves):

- Limited number of possible local interactions (precomputation of integrals to near machine precision)
- (plane wave) FMM for far-field
BOX CODES, IN BRIEF

Box codes (typically) work on level-restricted trees and are very efficient (density $f$ defined on leaves):

- Limited number of possible local interactions (precomputation of integrals to near machine precision)
- (plane wave) FMM for far-field
- Very fast, even on adaptive grids
CPU time vs. number of FMM nodes

- Adaptive grid
- Uniform grid
The application of a bounded operator is easy to analyze.
The application of a bounded operator is easy to analyze.

The box code computes the volume integral at **collocation nodes** to a specified precision.

**Notation:**
- $\tilde{f}$: approximation to $f$ by polynomials on each leaf
- $\tilde{V} \tilde{f}(x)$: value of $V \tilde{f}(x)$ computed using box code
- $\epsilon$: precision of FMM

From multipole estimates:

$$\left| \tilde{V} \tilde{f}(x) - Vf(x) \right| \leq \epsilon \| \tilde{f} \|_1,$$

From triangle inequality and boundedness of $V$:

$$\left| \tilde{V} \tilde{f}(x) - Vf(x) \right| \| \tilde{f} \|_\infty \leq \epsilon \vert \Omega \vert + C(\Omega) \| f - \tilde{f} \|_\infty \| \tilde{f} \|_\infty.$$

Gives an a priori error estimate (similar for $\nabla V$).
The application of a bounded operator is easy to analyze

The box code computes the volume integral at \textit{collocation nodes} to a specified precision.

Notation:
- $\tilde{f}$: approximation to $f$ by polynomials on each leaf
- $\tilde{V}\tilde{f}(x)$: value of $V\tilde{f}(x)$ computed using box code
- $\epsilon$: precision of FMM
The application of a bounded operator is easy to analyze.

The box code computes the volume integral at collocation nodes to a specified precision.

Notation:
- \( \tilde{f} \): approximation to \( f \) by polynomials on each leaf
- \( \tilde{V}\tilde{f}(x) \): value of \( V\tilde{f}(x) \) computed using box code
- \( \epsilon \): precision of FMM

From multipole estimates:

\[
|\tilde{V}\tilde{f}(x) - V\tilde{f}(x)| \leq \epsilon \|\tilde{f}\|_1 ,
\]
The application of a bounded operator is easy to analyze.

The box code computes the volume integral at collocation nodes to a specified precision.

Notation:
- \( \tilde{f} \): approximation to \( f \) by polynomials on each leaf
- \( \tilde{V}\tilde{f}(x) \): value of \( V\tilde{f}(x) \) computed using box code
- \( \epsilon \): precision of FMM

From multipole estimates:

\[
|\tilde{V}\tilde{f}(x) - V\tilde{f}(x)| \leq \epsilon \|\tilde{f}\|_1 ,
\]

From triangle inequality and boundedness of \( V \):

\[
\frac{|\tilde{V}\tilde{f}(x) - Vf(x)|}{\|\tilde{f}\|_\infty} \leq \epsilon |\Omega| + C(\Omega) \frac{\|f - \tilde{f}\|_\infty}{\|\tilde{f}\|_\infty} .
\]
The application of a bounded operator is easy to analyze.

The box code computes the volume integral at **collocation nodes** to a specified precision.

Notation:
- $\tilde{f}$: approximation to $f$ by polynomials on each leaf
- $\tilde{V}\tilde{f}(x)$: value of $V\tilde{f}(x)$ computed using box code
- $\epsilon$: precision of FMM

From multipole estimates:

$$|\tilde{V}\tilde{f}(x) - V\tilde{f}(x)| \leq \epsilon\|\tilde{f}\|_1,$$

From triangle inequality and boundedness of $V$:

$$\frac{|\tilde{V}\tilde{f}(x) - Vf(x)|}{\|\tilde{f}\|_\infty} \leq \epsilon|\Omega| + C(\Omega)\frac{\|f - \tilde{f}\|_\infty}{\|\tilde{f}\|_\infty}.$$ 

Gives an a priori error estimate (similar for $\nabla V$).
Let $\Omega$ be contained in a box $\Omega_B$ and let $f_e|_\Omega = f$ be defined on all of $\Omega_B$. Then

$$Vf_e(x) = \int_{\Omega_B} G_L(x, y)f_e(y) \, dy$$

is another particular solution and $Vf_e$ can be computed using a box code.

Figure: The domain $\Omega$ with an adaptive tree structure overlaying it.
What if a smooth extension $f_e$ is not readily available?

It must be computed in some way.

**Figure:** The domain $\Omega$ with an adaptive tree structure overlaying it.
Figure: Example of a “cut-cell”.

COMPUTING THE EXTENDED FUNCTION

Extend by zero

Local function extension

Global extension by layer potential

Globalized local extension

Ethridge and Greengard, 2001
Ethridge, 2000, Langston, 2012
Askham, 2016
Rachh and Askham, 2017
Fryklund et al., 2017 (PUX)
COMPUTING THE EXTENDED FUNCTION

- Extend by zero
  [Ethridge and Greengard, 2001]

**Figure:** Example of a “cut-cell”.

- Local function extension
  [Ethridge, 2000, Langston, 2012]

- Global extension by layer potential
  [Askham, 2016] \((C_0)\)

- Globalized local extension
  [Fryklund et al., 2017] \((C_1)\) and

- [Rachh and Askham, 2017]
Computing the Extended Function

- Extend by zero
  [Ethridge and Greengard, 2001]
- Local function extension
  [Ethridge, 2000, Langston, 2012]

**Figure:** Example of a “cut-cell”.

---

- Global extension by layer potential
  [Askham, 2016] ($C_0$)
- Globalized local extension
  [Rachh and Askham, 2017] ($C_1$)

---

**PUX**
COMPUTING THE EXTENDED FUNCTION

Figure: Example of a “cut-cell”.

- Extend by zero
  [Ethridge and Greengard, 2001]

- Local function extension
  [Ethridge, 2000, Langston, 2012]

- Global extension by layer potential
  [Askham, 2016] ($C^0$) and
  [Rachh and Askham, 2017] ($C^1$)
Computing the Extended Function

- Extend by zero
  [Ethridge and Greengard, 2001]

- Local function extension
  [Ethridge, 2000, Langston, 2012]

- Global extension by layer potential
  [Askham, 2016] ($C^0$) and
  [Rachh and Askham, 2017] ($C^1$)

- Globalized local extension
  [Fryklund et al., 2017] (PUX)

Figure: Example of a “cut-cell”. 
Let $f$ be defined on $\Omega$ with boundary $\Gamma$. Then, define a function $w$ on $\mathbb{R}^2 \setminus \Omega$ as the solution of

\[
\Delta w = 0 \quad \text{in} \quad \mathbb{R}^2 \setminus \Omega,
\]

\[
w = f|_\Gamma \quad \text{on} \quad \Gamma.
\]

Then $f_e = f$ on $\Omega$ and $f_e = w$ outside is a globally continuous extension of $f$. 
Let $f$ be defined on $\Omega$ with boundary $\Gamma$. Then, define a function $w$ on $\mathbb{R}^2 \setminus \Omega$ as the solution of

$$
\Delta w = 0 \quad \text{in} \quad \mathbb{R}^2 \setminus \Omega,
$$

$$
w = f|_{\Gamma} \quad \text{on} \quad \Gamma.
$$

Then $f_e = f$ on $\Omega$ and $f_e = w$ outside is a globally continuous extension of $f$.

- $w$ can be computed using the same numerical tools as for $u_h$ (generalized Gaussian quads, fast solvers, QBX)
Let $f$ be defined on $\Omega$ with boundary $\Gamma$. Then, define a function $w$ on $\mathbb{R}^2 \setminus \Omega$ as the solution of

$$
\Delta w = 0 \quad \text{in } \mathbb{R}^2 \setminus \Omega,
$$

$$
w = f|_\Gamma \quad \text{on } \Gamma.
$$

Then $f_e = f$ on $\Omega$ and $f_e = w$ outside is a globally continuous extension of $f$.

- $w$ can be computed using the same numerical tools as for $u_h$ (generalized Gaussian quads, fast solvers, QBX)
- Smoother extensions can be obtained as solutions of polyharmonic problems.
Recall the a priori error bound

\[
\frac{|\tilde{V}\tilde{f}_e(x) - Vf_e(x)|}{\|\tilde{f}_e\|_{\infty}} \leq \epsilon|\Omega| + C(\Omega) \frac{\|f_e - \tilde{f}_e\|_{\infty}}{\|\tilde{f}_e\|_{\infty}}
\]
Recall the a priori error bound

\[
\frac{|\tilde{V}\tilde{f}_e(x) - Vf_e(x)|}{\|\tilde{f}_e\|_\infty} \leq \epsilon|\Omega| + C(\Omega)\frac{\|f_e - \tilde{f}_e\|_\infty}{\|\tilde{f}_e\|_\infty}
\]

Implied convergence rate

<table>
<thead>
<tr>
<th></th>
<th>Conv. Order $Vf$</th>
<th>Conv. Order $\nabla Vf$</th>
</tr>
</thead>
<tbody>
<tr>
<td>zero extension</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C^0$ extension</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$C^1$ extension</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Recall the a priori error bound

\[
\frac{|\tilde{V}f_e(x) - Vf_e(x)|}{\|\tilde{f}_e\|_\infty} \leq \epsilon|\Omega| + C(\Omega) \frac{\|f_e - \tilde{f}_e\|_\infty}{\|\tilde{f}_e\|_\infty}
\]

Implied convergence rate

|                | Conv. Order | Conv. Order | \n|----------------|-------------|-------------|
| zero extension | 0           | 0           |
| \(C^0\) extension | 1           | 1           |
| \(C^1\) extension | 2           | 2           |

These aren’t amazing. What rate do we observe?
\[ \Delta u = f \quad \text{in} \quad \Omega , \]
\[ u = u_b \quad \text{on} \quad \Gamma . \]

We set \( f \) and \( u_b \) so that the solution \( u \) is given by
\[ u(x) = \sin(10(x_1 + x_2)) + x_1^2 - 3x_2 + 8 . \]
We extend $f$ using the method and tools described above.
CONVERGENCE RATE (UNIFORM GRID)

Error in potential

Error in gradient

How?
CONVERGENCE RATE (UNIFORM GRID)

Error in potential

Error in gradient

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Observed</th>
<th>Predicted</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td>zero extension</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>$C^0$ extension</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$C^1$ extension</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>
OBSERVED CONVERGENCE RATE

To see that you gain 1 order:

\[ v(x) = -\frac{1}{2\pi} \int \log \| x - y \| f(y) \, dy , \quad \nabla v(x) = -\frac{1}{2\pi} \int \frac{x - y}{\| x - y \|^2} f(y) \, dy \]
OBSERVED CONVERGENCE RATE

To see that you gain 1 order:

\[ v(x) = -\frac{1}{2\pi} \int \log \|x-y\| f(y) \, dy, \quad \nabla v(x) = -\frac{1}{2\pi} \int \frac{x-y}{\|x-y\|^2} f(y) \, dy \]

- Local contribution gets weighted by area of a cell (gain \( h^2 \) for \( \log r \) and \( h \) for \( 1/r \))
To see that you gain 1 order:

$$v(x) = -\frac{1}{2\pi} \int \log \|x - y\| f(y) \, dy,$$
$$\nabla v(x) = -\frac{1}{2\pi} \int \frac{x - y}{\|x - y\|^2} f(y) \, dy$$

- Local contribution gets weighted by area of a cell (gain $h^2$ for log $r$ and $h$ for $1/r$)
- For the far-field, only $O(1/h)$ of the boxes are irregular (have to add up carefully for gradient) and each is area $h^2$
OBSERVED CONVERGENCE RATE

To see that you gain 1 order:

\[ \nu(x) = -\frac{1}{2\pi} \int \log \|x-y\| f(y) \, dy, \quad \nabla \nu(x) = -\frac{1}{2\pi} \int \frac{x-y}{\|x-y\|^2} f(y) \, dy \]

- Local contribution gets weighted by area of a cell (gain \( h^2 \) for log \( r \) and \( h \) for \( 1/r \))
- For the far-field, only \( O(1/h) \) of the boxes are irregular (have to add up carefully for gradient) and each is area \( h^2 \)

The gain of 2 orders for \( u \) is somewhat mysterious!
What are good (a priori) strategies for adaptive grids? Recall that $\tilde{f}_e$ is the local polynomial interpolant on each box.
What are good (a priori) strategies for adaptive grids? Recall that $\tilde{f}_e$ is the local polynomial interpolant on each box.

1. Enforce that $\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
What are good (a priori) strategies for adaptive grids? Recall that $\tilde{f}_e$ is the local polynomial interpolant on each box.

1. Enforce that $\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
2. Enforce that $h^2\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
What are good (a priori) strategies for adaptive grids? Recall that $\tilde{f}_e$ is the local polynomial interpolant on each box.

1. Enforce that $\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
2. Enforce that $h^2\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
3. Enforce that $h\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
What are good (a priori) strategies for adaptive grids? Recall that $\tilde{f}_e$ is the local polynomial interpolant on each box.

1. Enforce that $\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
2. Enforce that $h^2\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
3. Enforce that $h\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
4. Hybrid: enforce one criterion on irregular boxes and another on regular boxes (these perform best)
What are good (a priori) strategies for adaptive grids? Recall that $\tilde{f}_e$ is the local polynomial interpolant on each box.

1 Enforce that $\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
2 Enforce that $h^2\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
3 Enforce that $h\|f_e - \tilde{f}_e\| \leq \text{tol}$ on each leaf
4 Hybrid: enforce one criterion on irregular boxes and another on regular boxes (these perform best)

Note that by storing local expansions and QBX expansions from a QBX FMM, the QBX method gives you an oracle for $f_e$
Results for hybrid schemes

- 1 on reg, $h$ on irreg
- 1 on reg, $h^2$ on irreg
- uniform

Adaptive Performance
MORE DIFFICULT PROBLEM

\[ \Delta u = f \quad \text{in } \Omega , \]
\[ u = u_b \quad \text{on } \Gamma . \]

We set \( f \) and \( u_b \) so that the solution \( u \) is given by

\[
    u(x) = \sin(10(x_1 + x_2)) + x_1^2 - 3x_2 + 8 + e^{-(500x_1)^2},
\]

which requires lots of refinement near the \( x_2 \) axis.
Figure: Error in potential vs. number of discretization nodes

Figure: Error in gradient vs. number of discretization nodes
FUTURE WORK

Some plans

- Apply modified biharmonic FMM to Navier-Stokes integral equation methods
- Release wrapped solver with latest and greatest QBX implementation
- Implement adaptive-friendly version of biharmonic code
Thank you.
*Integral-equation methods for inhomogeneous elliptic partial differential equations in complex geometry.*

A stabilized separation of variables method for the modified biharmonic equation.

An adaptive fast multipole accelerated poisson solver for complex geometries.
*Journal of Computational Physics, 344:1–22.*

The embedded boundary integral method for the incompressible navier-stokes equations.
*In Proceedings of the International Association for Boundary Element Methods 2002 Symposium.*

An adaptive fast solver for the modified helmholtz equation in two dimensions.

A new fast-multipole accelerated poisson solver in two dimensions.

*Fast algorithms for volume integrals in potential theory.*

Partition of unity extension of functions on complex domains.


