Solvers for Applications of Integral Equations

Adrianna Gillman

University of Colorado, Boulder

Collaborators:
Alex Barnett (Flatiron Institute)
Gary Marple (U of Michigan)
Per-Gunnar Martinsson (UT Austin)
Shravan Veerapaneni (U of Michigan)

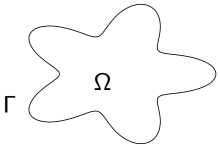
Former Student:
Yabin Zhang (U Michigan)

Conference on Fast Direct Solvers

October 23, 2021

Support provided by the NSF DMS-1522631, the Alfred P. Sloan Foundation, and TOTAL, Inc.

Model problem



Consider the problem

$$\begin{aligned} -\Delta u(\mathbf{x}) &= 0, \quad \mathbf{x} \in \Omega, \\ u(\mathbf{x}) &= g(\mathbf{x}), \quad \mathbf{x} \in \Gamma. \end{aligned}$$

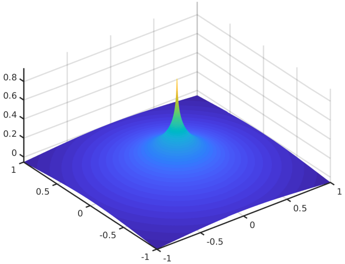
The union of physics and mathematics

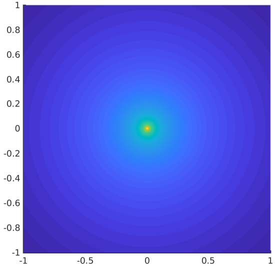
For a given point charge $\mathbf{x}_0 \in \mathbb{R}^2$, the solution of

$$-\Delta u(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_0), \quad \mathbf{x} \in \mathbb{R}^2$$

is

$$u(\mathbf{x}) = -\frac{1}{2\pi} \log |\mathbf{x} - \mathbf{x}_0|.$$





The union of physics and mathematics

For a given point charge $\mathbf{x}_0 \in \mathbb{R}^2$, the solution of

$$-\Delta u(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_0), \quad \mathbf{x} \in \mathbb{R}^2$$

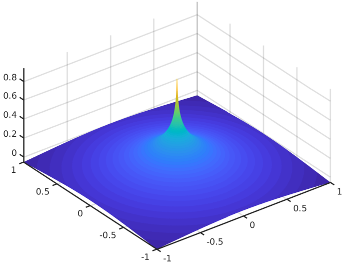
is

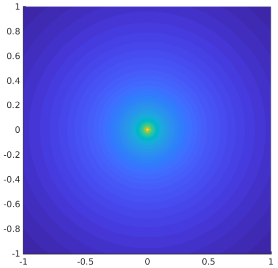
$$u(\mathbf{x}) = -\frac{1}{2\pi} \log |\mathbf{x} - \mathbf{x}_0|.$$

The fundamental solution $G(\mathbf{x}, \mathbf{y})$ is given by

$$G(\mathbf{x}, \mathbf{y}) = -\frac{1}{2\pi} \log |\mathbf{x} - \mathbf{y}|.$$

This allows us to move the point charge around.





The union of physics and mathematics

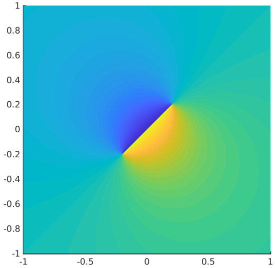
The double layer kernel

For a point **y** on the boundary of a curve, the double layer kernel

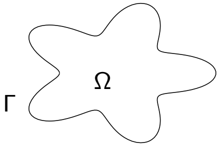
$$D(\mathbf{x},\mathbf{y}) = \partial_{\nu_{\mathbf{y}}} G(\mathbf{x},\mathbf{y})$$

is a solution of

$$-\Delta_{\mathbf{x}} u(\mathbf{x}) = \delta(\mathbf{x}-\mathbf{y}), \qquad \mathbf{x} \in \mathbb{R}^2.$$



Model problem



Consider the problem

$$-\Delta u(\mathbf{x}) = 0, \qquad \mathbf{x} \in \Omega,$$
$$u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \Gamma.$$

The solution can be represented as a double layer potential

$$u(\mathbf{x}) = \int_\Gamma \partial_{\nu_\mathbf{y}} G(\mathbf{x},\mathbf{y})\phi(\mathbf{y})ds(\mathbf{y}), \qquad \mathbf{x} \in \Omega,$$

where $\boldsymbol{\nu}_\mathbf{y}$ is the outward normal at $\mathbf{y}$ and $G(\mathbf{x},\mathbf{y})$ is the fundamental solution

$$G(\mathbf{x},\mathbf{y}) = -\frac{1}{2\pi}\log|\mathbf{x}-\mathbf{y}|.$$

Then the boundary charge distribution ϕ satisfies the boundary integral equation

$$\frac{1}{2}\phi(\mathbf{x}) + \int_\Gamma \partial_{\nu_\mathbf{y}} G(\mathbf{x},\mathbf{y})\phi(\mathbf{y})ds(\mathbf{y}) = g(\mathbf{x})$$

How do you discretize integral equations?

To discretize the integral equation using a Nyström method, pick an appropriate quadrature to approximate the integral. Then

$$g(\mathbf{x}) = \frac{1}{2}\phi(\mathbf{x}) + \int_\Gamma \partial_{\nu_\mathbf{y}} G(\mathbf{x},\mathbf{y})\phi(\mathbf{y})ds(\mathbf{y})$$
$$\sim \frac{1}{2}\phi(\mathbf{x}) + \sum_{j=1}^{N} \partial_{\nu_{\mathbf{x}_j}} G(\mathbf{x},\mathbf{x}_j)\phi(\mathbf{x}_j)w_j$$

Looking for the solution at the quadrature nodes and forcing the approximation to hold at these locations leads to a linear system where the i^{th} row is given by

$$g(\mathbf{x}_i) = \frac{1}{2}\phi(\mathbf{x}_i) + \sum_{j=1}^{N} \partial_{\nu_{\mathbf{x}_j}} G(\mathbf{x}_i,\mathbf{x}_j)\phi(\mathbf{x}_j)w_j$$

Model problem

Upon discretization, we have to solve a linear system of the form

$$\mathbf{A}\phi = (\frac{1}{2}\mathbf{I} + \mathbf{D})\phi = \mathbf{g},$$

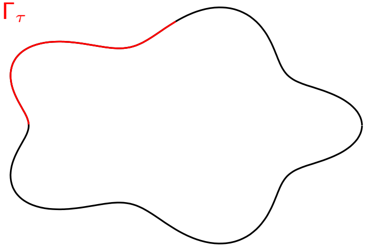
where $\mathbf{D}$ is a matrix that approximates the integral operator

$$\int_{\Gamma} \partial_{\nu_{\mathbf{y}}} G(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})ds(\mathbf{y}).$$

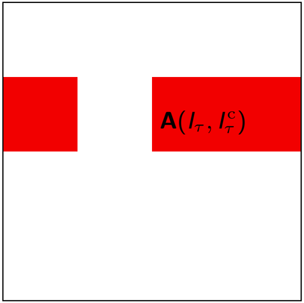
Properties of $\mathbf{A}$:

- Dense matrix.
- Size is determined by the number of discretization points.
- Data-sparse/structured matrix.

Is the BIE data sparse?



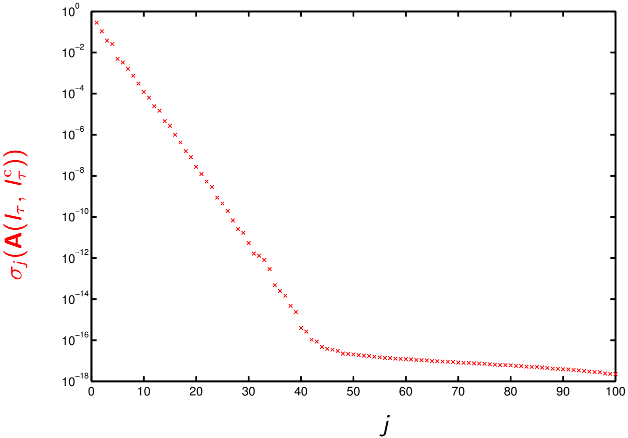
The contour Γ.



The matrix $\mathbf{A}$.

Is the BIE data sparse?

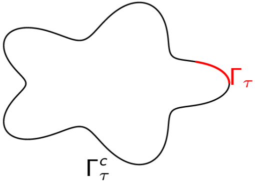
Singular values of $\mathbf{A}(I_\tau,\, I_\tau^c)$



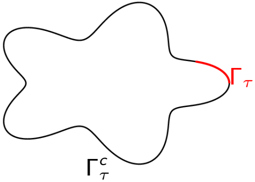
To precision 10^{-10}, the matrix $\mathbf{A}(I_\tau,\, I_\tau^c)$ has rank 29.

Fast factorization

Let N denote the number of discretization points on Γ. Suppose there are n_τ points on the Γ_τ.
Then there are $N - n_\tau$ points on Γ_τ^c.



Fast factorization



Let N denote the number of discretization points on Γ. Suppose there are n_τ points on the Γ_τ.
Then there are $N - n_\tau$ points on Γ_τ^c.

To create the low rank factorization of the off-diagonal block $\mathbf{A}_{\tau,\tau^c}$, the cost is

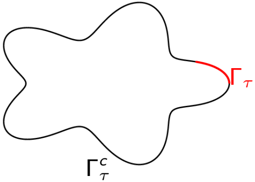
$$O(n_\tau^2(N - n_\tau)).$$

Thus the total cost of factoring via classic factorization techniques (QR, SVD, etc) over the entire geometry Γ is

$$O(N^3).$$

Fast factorization

Let N denote the number of discretization points on Γ. Suppose there are n_τ points on the Γ_τ.
Then there are $N - n_\tau$ points on Γ_τ^c.



To create the low rank factorization of the off-diagonal block $\mathbf{A}_{\tau,\tau^c}$, the cost is

$$O(n_\tau^2(N - n_\tau)).$$

Thus the total cost of factoring via classic factorization techniques (QR, SVD, etc) over the entire geometry Γ is

$$O(N^3).$$

By utilizing randomized linear algebra techniques such as rank revealing QR, the cost of factoring $\mathbf{A}_{\tau,\tau^c}$ is

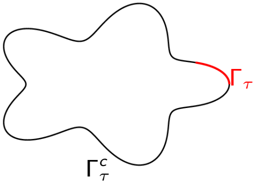
$$O(n_\tau(N - n_\tau)k)$$

where k is the ϵ-rank of the matrix and the total cost of factoring $\mathbf{A}$ is

$$O(N^2).$$

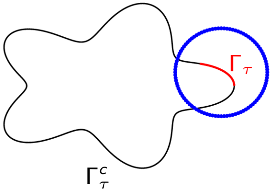
Fast factorization

This smoothness in the kernel away from the diagonal means that the interaction between far points can be represented to high accuracy via a collection of basis functions.



Fast factorization

Green's theorem says that for $\mathbf{x} \in \Gamma_{\tau^c}$ the field generated by charges on Γ_τ can be expressed by $\int_{\Gamma_p} G(\mathbf{x}, \mathbf{y})\sigma(\mathbf{y})dl(\mathbf{y})$.

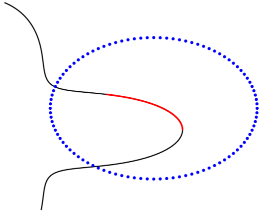


We can approximate this expression by

$$\sum_{j=1}^{n_{\text{proxy}}} c_j G(\mathbf{x}, \mathbf{x}_j)$$

where $\{\mathbf{x}_j\}_{j=1}^{n_{\text{proxy}}}$ are called proxy points and $\{c_j\}_{j=1}^{n_{\text{proxy}}}$ are coefficients that can be easily found (but are not needed).

Fast factorization



Green's theorem says that for $\mathbf{x} \in \Gamma_{\tau^c}$ the field generated by charges on Γ_τ can be expressed by $\int_{\Gamma_p} G(\mathbf{x}, \mathbf{y})\sigma(\mathbf{y})dl(\mathbf{y})$.

We can approximate this expression by

$$\sum_{j=1}^{n_{\rm proxy}} c_j G(\mathbf{x}, \mathbf{x}_j)$$

where $\{\mathbf{x}_j\}_{j=1}^{n_{\rm proxy}}$ are called proxy points and $\{c_j\}_{j=1}^{n_{\rm proxy}}$ are coefficients that can be easily found (but are not needed).

We can factor $[\mathbf{A}_{I_\tau, I_\tau^{\rm near}} | \mathbf{A}_{I_\tau, I_\tau^{\rm proxy}}]$ for a cost of

$$O(n_\tau(n_{\rm near} + n_{\rm proxy})k).$$

Thus computing the factorization has a $O(N)$ computational cost.

Numerical examples

These numerical examples were run on standard office desktops.

Most of the programs are written in Matlab (some in Fortran 77).

The reported CPU times have two components:

(1) Pre-computation (inversion, LU-factorization, constructing a Schur complement)

(2) Time for a single solve once pre-computation is completed

Numerical examples

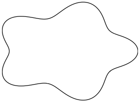
We invert a matrix approximating the operator

$$[A\,\phi](\mathbf{x}) = \frac{1}{2}\,\phi(\mathbf{x}) + \int_\Gamma D(\mathbf{x},\mathbf{y})\,\phi(\mathbf{y})\,ds(\mathbf{y}), \qquad \mathbf{x}\in\Gamma,$$

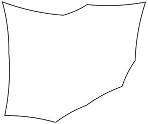
where D is the double layer kernel associated with Laplace's equation,

$$D(\mathbf{x},\mathbf{y}) = \frac{1}{2\pi}\,\frac{\mathbf{n}(\mathbf{y})\cdot(\mathbf{x}-\mathbf{y})}{|\mathbf{x}-\mathbf{y}|^2},$$

and where Γ is either one of the contours:



Smooth star

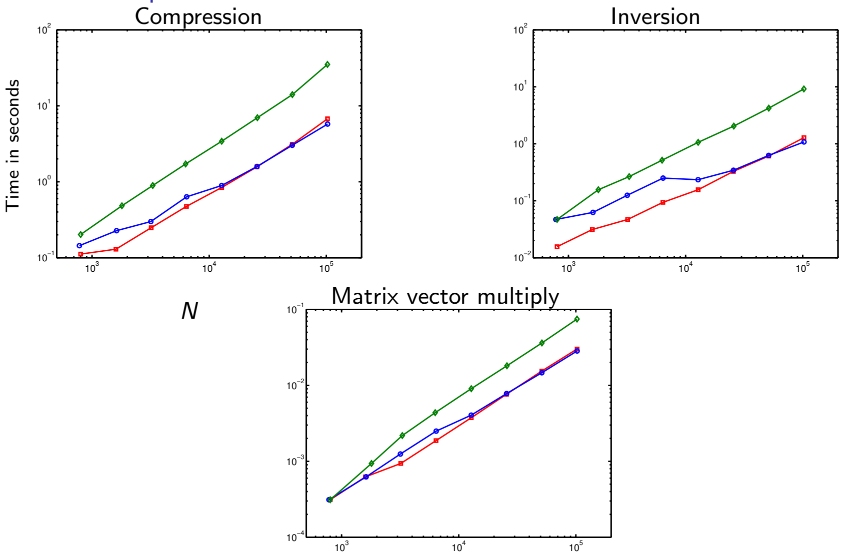


Star with corners
(local refinements at corners)

Snake
(# oscillations $\sim N$)

Examples from "A direct solver with $O(N)$ complexity for integral equations on one-dimensional domains," with P. Young, and P.G. Martinsson, 2012.

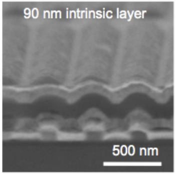
Numerical examples

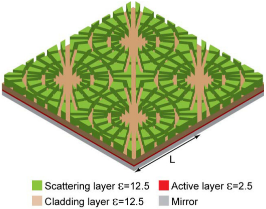


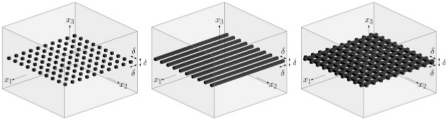
Within each graph, the three lines correspond to the three examples considered:

□ Smooth star ○ Star with corners ◇ Snake

Applications of involving periodic geometries



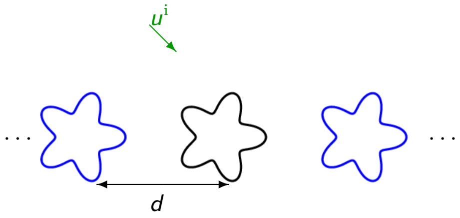




(a) Case (i) - discrete obstacles (b) Case (ii) - parallel wires (c) Case (iii) - wire mesh

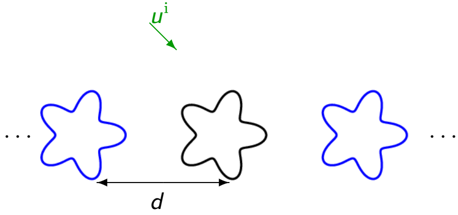
Figure 2: The domain Ω^δ in the three cases under consideration.

Definition of quasi-periodic scattering



- Let $\Omega \subset \mathbb{R}^2$ denote one obstacle. Then the collection of obstacles is expressed as $\Omega_{\mathbb{Z}} = \{\mathbf{x} : (x + nd, y) \in \Omega \text{ for some } n \in \mathbb{Z}\}$.
- The obstacles are hit by an incident plane wave $u^{\mathrm{i}} = e^{\mathrm{i}\mathbf{k}\cdot\mathbf{x}}$ where $|k| = \omega$.
- Our goal is to find the total field $u^{\mathrm{total}} = u^{\mathrm{inc}} + u$.
- Utilize the fact that each part of the field is quasi-periodic: i.e. $u(x + d, y) = \alpha u(x, y)$ where $\alpha = e^{\mathrm{i}\omega d \cos\theta}$ denotes the Bloch phase and θ is the angle of the incident wave.

Differential equation

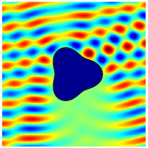


$$(\Delta + \omega^2) u(\mathbf{x}) = 0 \qquad \mathbf{x} \in \mathbb{R}^2 \setminus \Omega_{\mathbb{Z}}$$

$$u(\mathbf{x}) = -u^{\mathrm{i}}(\mathbf{x}) \qquad \mathbf{x} \in \partial \Omega_{\mathbb{Z}}$$

$$u \quad \text{`radiative'} \text{ as } \; y \to \pm\infty$$

Single object scattering



Consider the problem

$$(\Delta+\omega^2)u^{\mathrm{s}}(\mathbf{x})=0 \qquad \mathbf{x}\in\mathbb{R}\setminus\Omega$$
$$u^{\mathrm{s}}(\mathbf{x})=u^{\mathrm{i}}(\mathbf{x}) \qquad \mathbf{x}\in\partial\Omega$$
$$u^{\mathrm{s}} \quad \text{'radiative' far from } \Omega$$

The solution can be represented as a double layer potential

$$u^{\mathrm{s}}(\mathbf{x})=\int_\Gamma \partial_{\boldsymbol{\nu}} G_\omega(\mathbf{x},\mathbf{y})\tau(\mathbf{y})ds(\mathbf{y}), \qquad \mathbf{x}\in\Omega,$$

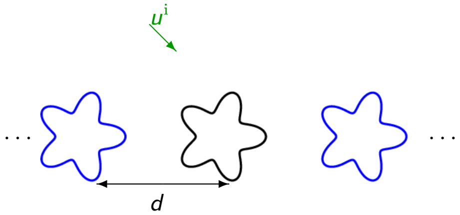
where $\boldsymbol{\nu}$ is the outward normal and $G_\omega(\mathbf{x},\mathbf{y})$ is the fundamental solution

$$G_\omega(\mathbf{x},\mathbf{y})=\frac{i}{4}H_0^{(1)}(\omega|\mathbf{x}-\mathbf{y}|).$$

Then the boundary charge distribution τ satisfies the boundary integral equation

$$-\frac{1}{2}\tau(\mathbf{x})+\int_\Gamma \partial_{\boldsymbol{\nu}} G_\omega(\mathbf{x},\mathbf{y})\tau(\mathbf{y})ds(\mathbf{y})=u^{\mathrm{i}}(\mathbf{x})$$

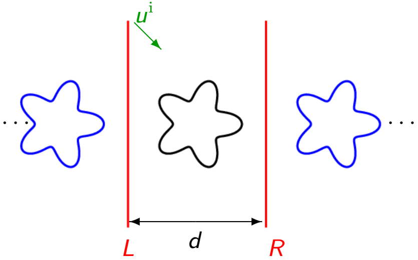
The standard way



Use the same integral equation but replace $G_\omega(\mathbf{x}, \mathbf{y})$ by
$G_{\omega,\mathrm{QP}}(\mathbf{x}) := \sum_{m\in\mathbb{Z}} \alpha^m G_\omega(\mathbf{x} - md)$ where α is the Bloch phase.

This has some problems...

One approach to solving the periodic problem



Let the solution be represented as a double layer potential plus a quasi-periodic potential

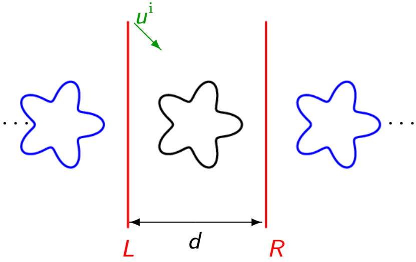
$$u(\mathbf{x}) = \sum_{j=-1}^{1} \alpha^j \int_{\partial\Omega} \partial_{\boldsymbol{\nu}} G_\omega(\mathbf{x}, \mathbf{y} + j\mathbf{d})\tau(\mathbf{y})ds(\mathbf{y}) + u_{QP}[\xi].$$

New condition: vanishing 'discrepancy'

$$\begin{cases} u_L - \alpha^{-1} u_R &= 0 \\ u_{nL} - \alpha^{-1} u_{nR} &= 0 \end{cases}$$

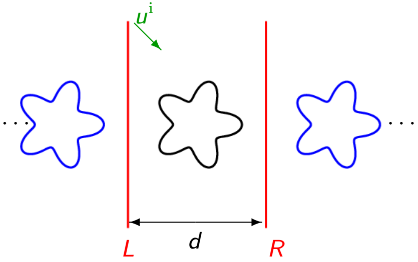
L. Greengard and A. Barnett (2011)

One approach to solving the periodic problem



$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \tau \\ \xi \end{bmatrix} = \begin{bmatrix} -\mathbf{u}^{\mathrm{i}} \\ \mathbf{0} \end{bmatrix}$$

One approach to solving the periodic problem



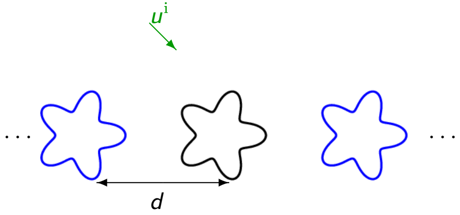
$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \tau \\ \xi \end{bmatrix} = \begin{bmatrix} -\mathbf{u}^{\mathrm{i}} \\ \mathbf{0} \end{bmatrix}$$

Instead of computing a pseudo-inverse of the matrix, we can compute the solution via a 2×2 block solve.

$$\xi = (\mathbf{Q} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{A}^{-1}\mathbf{u}^{\mathrm{inc}}$$
$$\tau = \mathbf{A}^{-1}\mathbf{u}^{\mathrm{inc}} - \mathbf{A}^{-1}\mathbf{B}\xi$$

For a problem with a fixed wave number ω, many incident waves will share a Bloch phase so the inversion technique can be reused. Cost: $O(N + M^3)$

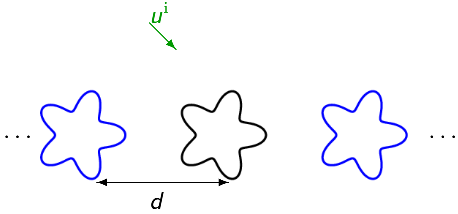
A faster direct solver for quasi-periodic scattering



When more than one Bloch phase α is of interest, it is possible to save more computational cost by splitting up the factors.

Recall: $\mathbf{A} = \mathbf{A}_0 + \alpha^{-1}\mathbf{A}_{-1} + \alpha\mathbf{A}_1.$

A faster direct solver for quasi-periodic scattering



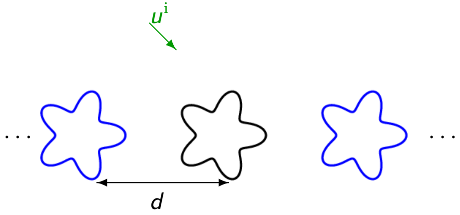
When more than one Bloch phase α is of interest, it is possible to save more computational cost by splitting up the factors.

Recall: $\mathbf{A} = \mathbf{A}_0 + \alpha^{-1}\mathbf{A}_{-1} + \alpha\mathbf{A}_1$.

$\mathbf{A}_{-1}$ and $\mathbf{A}_1$ only depend on wave number ω and are low rank.

Thus $\mathbf{A} = \mathbf{A}_0 + \mathbf{LR}$ where $\mathbf{L}$ and $\mathbf{R}$ are of size $N \times k$, $k \ll N$ and the factorizations need only be computed once independent of the incident wave.

A faster direct solver for quasi-periodic scattering



When more than one Bloch phase α is of interest, it is possible to save more computational cost by splitting up the factors.

Recall: $\mathbf{A} = \mathbf{A}_0 + \alpha^{-1}\mathbf{A}_{-1} + \alpha\mathbf{A}_1$.

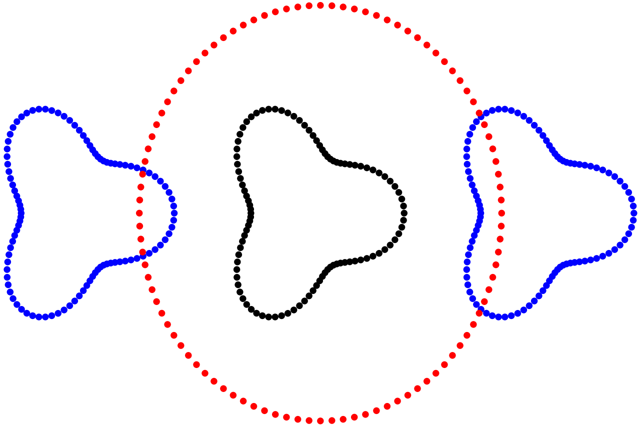
$\mathbf{A}_{-1}$ and $\mathbf{A}_1$ only depend on wave number ω and are low rank.

Thus $\mathbf{A} = \mathbf{A}_0 + \mathbf{LR}$ where $\mathbf{L}$ and $\mathbf{R}$ are of size $N \times k$, $k \ll N$ and the factorizations need only be computed once independent of the incident wave.

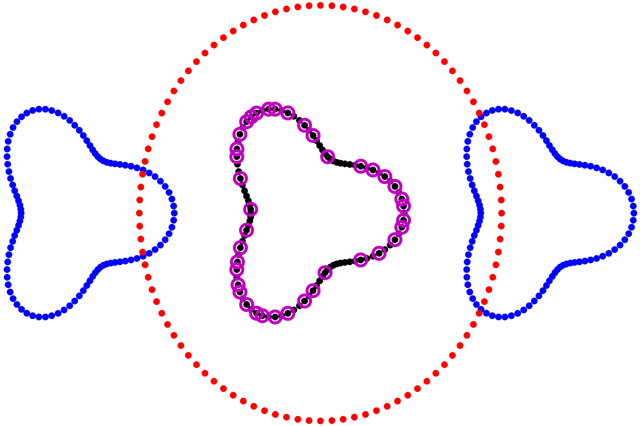
Woodbury Formula: inversion of a low-rank update

$$(\mathbf{A}_0 + \hat{\mathbf{A}})^{-1} = (\mathbf{A}_0 + \mathbf{LR})^{-1} = \mathbf{A}_0^{-1} - \mathbf{A}_0^{-1}\mathbf{L}\left(\mathbf{I} + \mathbf{R}\mathbf{A}_0^{-1}\mathbf{L}\right)^{-1}\mathbf{R}\mathbf{A}_0^{-1}$$

Compression with neighbors



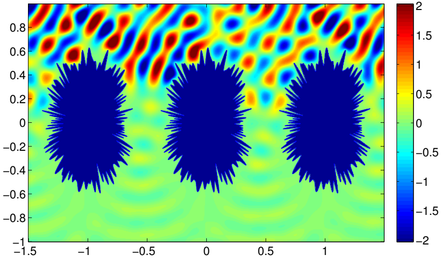
Compression with neighbors



There are 39 skeleton points.

Example

Multiple incident waves



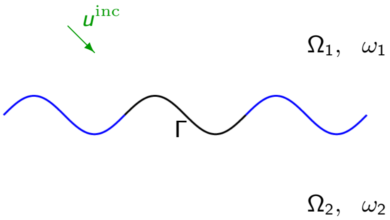
The FMM + GMRES takes one hour (248 iterations) to solve for the densities for one incident wave.

The fast direct solver takes 19.1 minutes to solve 200 densities.

(4.1 minutes of precomputation and 15 minutes for the block solves.)

Example from "A fast direct solver for quasi-periodic scattering problems," with A. Barnett, 2013.

One interface problem

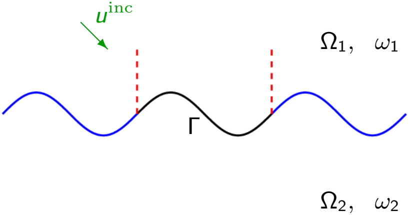


Consider the problem

$$
\begin{aligned}
(\Delta + \omega_1^2)u_1(\mathbf{x}) &= 0 & \mathbf{x} \in \Omega_1 \\
(\Delta + \omega_2^2)u_2(\mathbf{x}) &= 0 & \mathbf{x} \in \Omega_2 \\
u_1 - u_2 &= -u^{\text{inc}}(\mathbf{x}) & \mathbf{x} \in \Gamma \\
\frac{\partial u_1}{\partial \nu} - \frac{\partial u_2}{\partial \nu} &= -\frac{\partial u^{\text{inc}}}{\partial \nu} & \mathbf{x} \in \Gamma
\end{aligned}
$$

(Cho and Barnett, 2015)

One interface problem



Consider the problem

$$
\begin{aligned}
(\Delta + \omega_1^2) u_1(\mathbf{x}) &= 0 && \mathbf{x} \in \Omega_1 \\
(\Delta + \omega_2^2) u_2(\mathbf{x}) &= 0 && \mathbf{x} \in \Omega_2 \\
u_1 - u_2 &= -u^{\mathrm{inc}}(\mathbf{x}) && \mathbf{x} \in \Gamma \\
\frac{\partial u_1}{\partial \nu} - \frac{\partial u_2}{\partial \nu} &= -\frac{\partial u^{\mathrm{inc}}}{\partial \nu} && \mathbf{x} \in \Gamma
\end{aligned}
$$

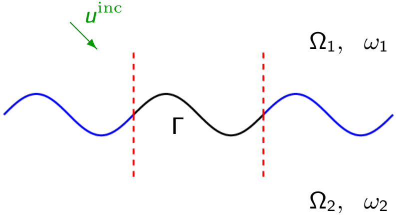
We represent the solution in Ω_1 by

$$
u_1(\mathbf{x}) = \sum_{j=-1}^{1} \alpha^j \int_\Gamma \partial_{\boldsymbol{\nu}} G_{\omega_1}(\mathbf{x}, \mathbf{y} + j\mathbf{d}) \tau(\mathbf{y}) dl(\mathbf{y}) + \sum_{j=-1}^{1} \alpha^j \int_\Gamma G_{\omega_1}(\mathbf{x}, \mathbf{y} + j\mathbf{d}) \sigma(\mathbf{y}) dl(\mathbf{y})
$$

$$
+ u_{QP}^1[c].
$$

(Cho and Barnett, 2015)

One interface problem



Consider the problem

$$\begin{aligned}
(\Delta + \omega_1^2) u_1(\mathbf{x}) &= 0 && \mathbf{x} \in \Omega_1 \\
(\Delta + \omega_2^2) u_2(\mathbf{x}) &= 0 && \mathbf{x} \in \Omega_2 \\
u_1 - u_2 &= -u^{\text{inc}}(\mathbf{x}) && \mathbf{x} \in \Gamma \\
\frac{\partial u_1}{\partial \nu} - \frac{\partial u_2}{\partial \nu} &= -\frac{\partial u^{\text{inc}}}{\partial \nu} && \mathbf{x} \in \Gamma
\end{aligned}$$

We represent the solution in Ω_1 by

$$u_1(\mathbf{x}) = \sum_{j=-1}^{1} \alpha^j \int_\Gamma \partial_{\boldsymbol{\nu}} G_{\omega_1}(\mathbf{x}, \mathbf{y} + j\mathbf{d}) \tau_1(\mathbf{y}) dl(\mathbf{y}) + \sum_{j=-1}^{1} \alpha^j \int_\Gamma G_{\omega_1}(\mathbf{x}, \mathbf{y} + j\mathbf{d}) \sigma_1(\mathbf{y}) dl(\mathbf{y})$$

$$+ u_{QP}^1[c].$$

Likewise, we represent the solution in Ω_2 by

$$u_2(\mathbf{x}) = \sum_{j=-1}^{1} \alpha^j \int_\Gamma \partial_{\boldsymbol{\nu}} G_{\omega_2}(\mathbf{x}, \mathbf{y} + j\mathbf{d}) \tau_2(\mathbf{y}) dl(\mathbf{y}) + \sum_{j=-1}^{1} \alpha^j \int_\Gamma G_{\omega_2}(\mathbf{x}, \mathbf{y} + j\mathbf{d}) \sigma_2(\mathbf{y}) dl(\mathbf{y})$$

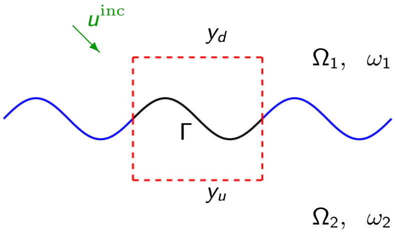
$$+ u_{QP}^2[c].$$

(Cho and Barnett, 2015)

One interface problem

Consider the problem

$$(\Delta + \omega_1^2) u_1(\mathbf{x}) = 0 \qquad \mathbf{x} \in \Omega_1$$
$$(\Delta + \omega_2^2) u_2(\mathbf{x}) = 0 \qquad \mathbf{x} \in \Omega_2$$
$$u_1 - u_2 = -u^{\mathrm{inc}}(\mathbf{x}) \qquad \mathbf{x} \in \Gamma$$
$$\frac{\partial u_1}{\partial \nu} - \frac{\partial u_2}{\partial \nu} = -\frac{\partial u^{\mathrm{inc}}}{\partial \nu} \qquad \mathbf{x} \in \Gamma$$



The radiation condition for $y > y_u$ can be characterized by the uniform convergence of the Rayleigh-Bloch expansion in the upper half-space

$$u(x,y) = \sum_{n \in \mathbb{Z}} a_n e^{\mathrm{i}\kappa_n x} e^{\mathrm{i}k_n (y - y_u)} \tag{1}$$

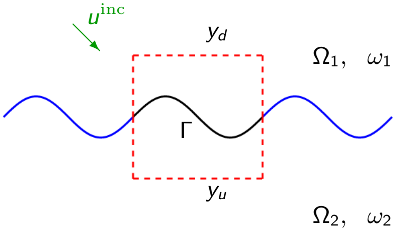
where $\kappa_n := \omega_1 \cos\theta^{\mathrm{inc}} + \frac{2\pi n}{d}$ and $k_n = \sqrt{\omega_1^2 - \kappa_n^2}$.

(Cho and Barnett, 2015)

One interface problem

Consider the problem

$$
\begin{aligned}
(\Delta + \omega_1^2)u_1(\mathbf{x}) &= 0 && \mathbf{x} \in \Omega_1\\
(\Delta + \omega_2^2)u_2(\mathbf{x}) &= 0 && \mathbf{x} \in \Omega_2\\
u_1 - u_2 &= -u^{\mathrm{inc}}(\mathbf{x}) && \mathbf{x} \in \Gamma\\
\frac{\partial u_1}{\partial \nu} - \frac{\partial u_2}{\partial \nu} &= -\frac{\partial u^{\mathrm{inc}}}{\partial \nu} && \mathbf{x} \in \Gamma
\end{aligned}
$$



After enforcing matching conditions for the expansions and the quasi-periodic boundary conditions, we are left solving the following linear system.

$$
\begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{C} & \mathbf{Q} & \mathbf{0} \\ \mathbf{Z} & \mathbf{V} & \mathbf{W} \end{bmatrix}
\begin{bmatrix} \hat{\boldsymbol{\sigma}} \\ \mathbf{c} \\ \mathbf{a} \end{bmatrix}
=
\begin{bmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}
$$

where $\hat{\boldsymbol{\sigma}} = \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{\sigma} \end{bmatrix}$ and $\mathbf{f} = \begin{bmatrix} -\mathbf{u}^{\mathrm{inc}} \\ -\frac{\partial \mathbf{u}^{\mathrm{inc}}}{\partial \nu} \end{bmatrix}$.

A closer look at $\mathbf{A}$

The matrix $\mathbf{A}$ is given by the following expression

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} + \tilde{\mathbf{D}}_1 - \tilde{\mathbf{D}}_2 & \tilde{\mathbf{S}}_1 - \tilde{\mathbf{S}}_2 \\ \tilde{\mathbf{T}}_1 - \tilde{\mathbf{T}}_2 & -\mathbf{I} + \tilde{\mathbf{D}}_1^* - \tilde{\mathbf{D}}_2^* \end{bmatrix}$$

where $\tilde{\mathbf{D}}_1 - \tilde{\mathbf{D}}_2$ denotes the discretized integral operators

$$\sum_{j=-1}^{1} \alpha^j \int_\Gamma \partial_{\boldsymbol{\nu}} G_{\omega_1}(\mathbf{x}, \mathbf{y} + j\mathbf{d}) \tau(\mathbf{y}) dl(\mathbf{y}) - \sum_{j=-1}^{1} \alpha^j \int_\Gamma \partial_{\boldsymbol{\nu}} G_{\omega_2}(\mathbf{x}, \mathbf{y} + j\mathbf{d}) \tau(\mathbf{y}) dl(\mathbf{y}),$$

$\tilde{\mathbf{S}}_1 - \tilde{\mathbf{S}}_2$ denotes the discretized integral operators

$$\sum_{j=-1}^{1} \alpha^j \int_\Gamma G_{\omega_1}(\mathbf{x}, \mathbf{y} + j\mathbf{d}) \sigma(\mathbf{y}) dl(\mathbf{y}) - \sum_{j=-1}^{1} \alpha^j \int_\Gamma G_{\omega_2}(\mathbf{x}, \mathbf{y} + j\mathbf{d}) \sigma(\mathbf{y}) dl(\mathbf{y}), \ldots$$

The block solve

We chose to find the solution to the linear system

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{C} & \mathbf{Q} & \mathbf{0} \\ \mathbf{Z} & \mathbf{V} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\sigma}} \\ \mathbf{c} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

via a similar block solve. In other words,

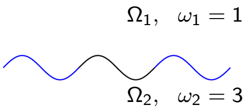
$$\hat{\boldsymbol{\sigma}} = -\mathbf{A}^{-1}[\mathbf{B} \;\; \mathbf{0}] \begin{bmatrix} \mathbf{c} \\ \mathbf{a} \end{bmatrix} + \mathbf{A}^{-1}\mathbf{f}$$

and

$$\begin{bmatrix} \mathbf{c} \\ \mathbf{a} \end{bmatrix} = -\left(\begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{V} & \mathbf{W} \end{bmatrix} - \begin{bmatrix} \mathbf{C} \\ \mathbf{Z} \end{bmatrix} \mathbf{A}^{-1}[\mathbf{B} \;\; \mathbf{0}] \right)^{\dagger} \begin{bmatrix} \mathbf{C} \\ \mathbf{Z} \end{bmatrix} \mathbf{A}^{-1}\mathbf{f}.$$

As with the original solution technique, the matrix $\mathbf{A} = \mathbf{A}_0 + \alpha^{-1}\mathbf{A}_{-1} + \alpha\mathbf{A}_1$ will be broken up so that its factors can be reused for all choices of Bloch phase α.

Numerical example



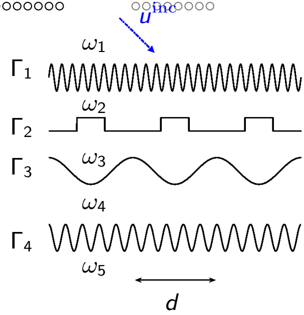
Accuracy of compression set to 10^{-10}

N	1584	3184	6384	12784	25584	51184
T_{build}	18.60	36.45	68.50	130.42	250.57	482.01
T_{apply}	0.43	0.81	1.64	3.72	10.96	19.64

Multi-layered medium problem

Consider the problem

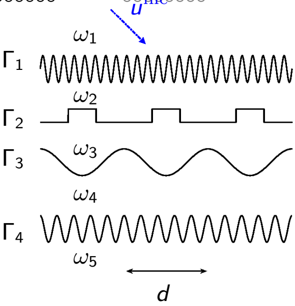
$$(\Delta + \omega_i^2) u_i(\mathbf{x}) = 0 \qquad \mathbf{x} \in \Omega_i$$
$$u_1 - u_2 = -u^{\mathrm{inc}}(\mathbf{x}) \qquad \mathbf{x} \in \Gamma_1$$
$$\frac{\partial u_1}{\partial \nu} - \frac{\partial u_2}{\partial \nu} = -\frac{\partial u^{\mathrm{inc}}}{\partial \nu} \qquad \mathbf{x} \in \Gamma_1$$
$$u_i - u_{i+1} = 0 \qquad \mathbf{x} \in \Gamma_i,\ i > 1$$
$$\frac{\partial u_i}{\partial \nu} - \frac{\partial u_{i+1}}{\partial \nu} = 0 \qquad \mathbf{x} \in \Gamma_i,\ i > 1$$



Multi-layered medium problem

Consider the problem

$$
\begin{aligned}
&(\Delta + \omega_i^2) u_i(\mathbf{x}) = 0 && \mathbf{x} \in \Omega_i \\
&u_1 - u_2 = -u^{\mathrm{inc}}(\mathbf{x}) && \mathbf{x} \in \Gamma_1 \\
&\frac{\partial u_1}{\partial \nu} - \frac{\partial u_2}{\partial \nu} = -\frac{\partial u^{\mathrm{inc}}}{\partial \nu} && \mathbf{x} \in \Gamma_1 \\
&u_i - u_{i+1} = 0 && \mathbf{x} \in \Gamma_i,\ i > 1 \\
&\frac{\partial u_i}{\partial \nu} - \frac{\partial u_{i+1}}{\partial \nu} = 0 && \mathbf{x} \in \Gamma_i,\ i > 1
\end{aligned}
$$

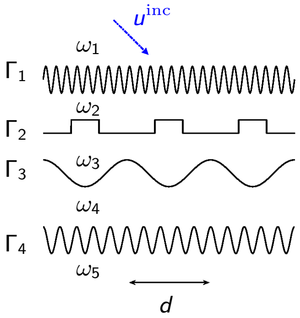


The solution on the top and bottoms are as before. The solution for the middle layers given expressed as

$$
\begin{aligned}
u_i(\boldsymbol{x}) = &\sum_{j=-1}^{1} \alpha^j \int_{\Gamma_i} \partial_{\nu_{\mathbf{y}}} G_{\omega_i}(\mathbf{x}, \mathbf{y} + jd) \sigma_i(\mathbf{y}) dl(\mathbf{y}) + \sum_{j=-1}^{1} \alpha^j \int_{\Gamma_i} G_{\omega_i}(\mathbf{x}, \mathbf{y} + jd) \tau_i(\mathbf{y}) dl(\mathbf{y}) \\
&+ \sum_{j=-1}^{1} \alpha^j \int_{\Gamma_{i-1}} \partial_{\nu_{\mathbf{y}}} G_{\omega_i}(\mathbf{x}, \mathbf{y} + jd) \sigma_{i-1}(\mathbf{y}) dl(\mathbf{y}) \\
&+ \sum_{l=-1}^{1} \alpha^j \int_{\Gamma_{i-1}} G_{\omega_i}(\mathbf{x}, \mathbf{y} + jd) \tau_{i-1}(\mathbf{y}) dl(\mathbf{y}) + u_{QP}^i[c_i]
\end{aligned}
$$

Multi-layered medium problem

Upon enforcing boundary conditions, periodizing conditions and radiation conditions, one has to solve a block system of the form



$$\begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{C} & \mathbf{Q} & \mathbf{0} \\ \mathbf{Z} & \mathbf{V} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\sigma}} \\ \mathbf{c} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

where $\hat{\boldsymbol{\sigma}} = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\sigma}_1 \\ \boldsymbol{\tau}_2 \\ \boldsymbol{\sigma}_2 \\ \boldsymbol{\tau}_3 \\ \boldsymbol{\sigma}_3 \\ \boldsymbol{\tau}_4 \\ \boldsymbol{\sigma}_4 \end{bmatrix}$ $\mathbf{f} = \begin{bmatrix} -\mathbf{u}^{\text{inc}} \\ -\frac{\partial \mathbf{u}^{\text{inc}}}{\partial \nu} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$ and $\mathbf{A}$ is a block tridiagonal matrix.

The proposed solver

We chose to find the solution to the linear system

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{0} \\ \mathbf{C} & \mathbf{Q} & \mathbf{0} \\ \mathbf{Z} & \mathbf{V} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\sigma}} \\ \mathbf{c} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

via a block solve. In other words,

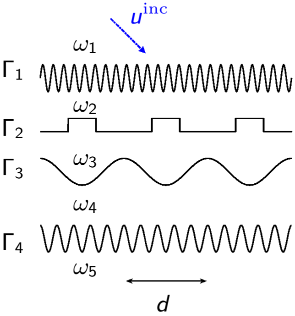
$$\hat{\boldsymbol{\sigma}} = -\mathbf{A}^{-1}[\mathbf{B} \;\; \mathbf{0}] \begin{bmatrix} \mathbf{c} \\ \mathbf{a} \end{bmatrix} + \mathbf{A}^{-1}\mathbf{f}$$

and

$$\begin{bmatrix} \mathbf{c} \\ \mathbf{a} \end{bmatrix} = -\left(\begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{V} & \mathbf{W} \end{bmatrix} - \begin{bmatrix} \mathbf{C} \\ \mathbf{Z} \end{bmatrix} \mathbf{A}^{-1}[\mathbf{B} \;\; \mathbf{0}] \right)^{\dagger} \begin{bmatrix} \mathbf{C} \\ \mathbf{Z} \end{bmatrix} \mathbf{A}^{-1}\mathbf{f}.$$

Goal: Construct a fast direct solver for the matrix **A** where the precomputation can be used independent of Bloch phase α.

A closer look at $\mathbf{A}$



The matrix $\mathbf{A}$ has the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{32} & \mathbf{A}_{33} & \mathbf{A}_{34} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{43} & \mathbf{A}_{44} \end{bmatrix}$$

where $\mathbf{A}_{11} = \mathbf{A}_{0,11} + \alpha^{-1}\mathbf{A}_{-1,11} + \alpha\mathbf{A}_{1,11}$, $\mathbf{A}_{12} = \mathbf{A}_{0,12} + \alpha^{-1}\mathbf{A}_{-1,12} + \alpha\mathbf{A}_{1,12}$, etc.

Fast application of the inverse of $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{0,11} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{0,22} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{0,33} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{0,44} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{pm,11} & \mathbf{A}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{pm,22} & \mathbf{A}_{23} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{32} & \mathbf{A}_{pm,33} & \mathbf{A}_{34} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{43} & \mathbf{A}_{pm,44} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{A}_{0,11} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{0,22} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{0,33} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{0,44} \end{bmatrix} + \begin{bmatrix} \mathbf{L}_{11}\mathbf{R}_{11} & \mathbf{L}_{12}\mathbf{R}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{21}\mathbf{R}_{21} & \mathbf{L}_{22}\mathbf{R}_{22} & \mathbf{L}_{23}\mathbf{R}_{23} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{32}\mathbf{R}_{32} & \mathbf{L}_{33}\mathbf{R}_{33} & \mathbf{L}_{34}\mathbf{R}_{34} \\ \mathbf{0} & \mathbf{0} & \mathbf{L}_{43}\mathbf{R}_{43} & \mathbf{L}_{44}\mathbf{R}_{44} \end{bmatrix}$$

where $\mathbf{A}_{pm,11} = \alpha^{-1}\mathbf{A}_{-1,11} + \alpha\mathbf{A}_{1,11}$, $\mathbf{A}_{12} = \mathbf{A}_{0,12} + \alpha^{-1}\mathbf{A}_{-1,12} + \alpha\mathbf{A}_{1,12}$,etc.

Fast application of the inverse of $\mathbf{A}$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{0,11} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{0,22} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{0,33} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{0,44} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{pm,11} & \mathbf{A}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{21} & \mathbf{A}_{pm,22} & \mathbf{A}_{23} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{32} & \mathbf{A}_{pm,33} & \mathbf{A}_{34} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{43} & \mathbf{A}_{pm,44} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{A}_{0,11} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{0,22} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{0,33} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_{0,44} \end{bmatrix} + \begin{bmatrix} \mathbf{L}_{11}\mathbf{R}_{11} & \mathbf{L}_{12}\mathbf{R}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_{21}\mathbf{R}_{21} & \mathbf{L}_{22}\mathbf{R}_{22} & \mathbf{L}_{23}\mathbf{R}_{23} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{32}\mathbf{R}_{32} & \mathbf{L}_{33}\mathbf{R}_{33} & \mathbf{L}_{34}\mathbf{R}_{34} \\ \mathbf{0} & \mathbf{0} & \mathbf{L}_{43}\mathbf{R}_{43} & \mathbf{L}_{44}\mathbf{R}_{44} \end{bmatrix}$$

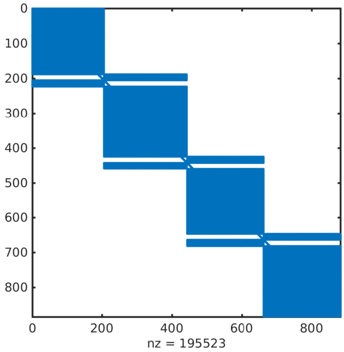
where $\mathbf{A}_{pm,11} = \alpha^{-1}\mathbf{A}_{-1,11} + \alpha\mathbf{A}_{1,11}$, $\mathbf{A}_{12} = \mathbf{A}_{0,12} + \alpha^{-1}\mathbf{A}_{-1,12} + \alpha\mathbf{A}_{1,12}$,etc.

Recall: **Woodbury Formula:** inversion of a low-rank update

$$(\mathbf{A}_0 + \hat{\mathbf{A}})^{-1} = (\mathbf{A}_0 + \mathbf{L}\mathbf{R})^{-1} = \mathbf{A}_0^{-1} - \mathbf{A}_0^{-1}\mathbf{L}\left(\mathbf{I} + \mathbf{R}\mathbf{A}_0^{-1}\mathbf{L}\right)^{-1}\mathbf{R}\mathbf{A}_0^{-1}$$

Sparsity pattern of $(\mathbf{I} + \mathbf{R}\mathbf{A}^{-1}\mathbf{L})$

$N_l = 608$ points per level



Numerical experiments

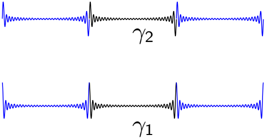
The algorithm was implemented in Matlab and ran on a desktop computer.

The computational cost of the direct solution technique can be broken into 4 parts:

- Precomputation I: The fast precomputation that is independent of Bloch phase α. (Factors for $\mathbf{A}$)
- Precomputation II: The precomputation that is independent of Bloch phase α but not fast. ($\mathbf{B}$, $\mathbf{C}$, $\mathbf{Z}$ and $\mathbf{Q}$)
- Precomputation III: The precomputation that depends on the incident angle only through dependence on the Bloch phase α. (Scaling matrices by Bloch phase, construct FDS for $\mathbf{A}^{-1}$, $\mathbf{A}^{-1}\mathbf{B}$, evaluating $\mathbf{W}$, evaluating the Schur complement, computing the SVD)
- Solve: The evaluations that are dependent on the incident wave $\theta^{\rm inc}$. (Evaluate $\mathbf{A}^{-1}\mathbf{f}$ and solve for the unknowns $\hat{\boldsymbol{\sigma}}$, $\mathbf{c}$ and $\mathbf{a}$.)

Nine layers

ω alternates between 40 and $40\sqrt{2}$



Accuracy of compression: 10^{-12}

N_i	T_I	T_{II}	T_{III}	T_{solve}	Flux error
1280	201	3.8	14.9	0.6	2.1e-4
2560	407	7.6	31.5	2.5	1.2e-5
5120	768	13.4	61.5	4.9	1.5e-7
10240	1390	26.1	117.1	13.5	4.6e-11
20480	2360	51.4	231.2	27.9	4.6e-10

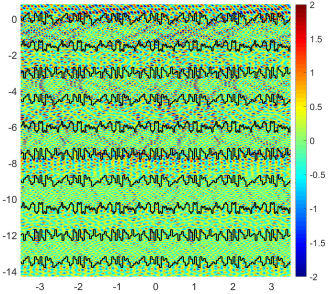
Example from "A fast direct solution technique for quasi-periodic scattering in multi-layered medium," with Y. Zhang.

In action on an eleven layer geometry

287 different angles and 24 independent Bloch phases

$$N = 121136$$

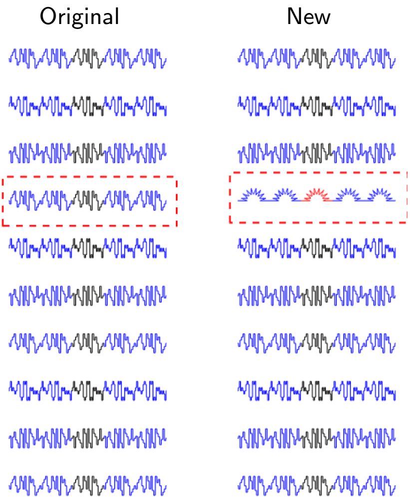
	(sec)
T_I	2369.3
T_{II}	32.3
T_{III}	4517.4 $\forall\alpha$
	188.2 per Bloch phase
T_{solve}	482.2 $\forall\theta$
	(1.7 per incident angle)



Times for building the solver for one incident angle θ.
T_I: 237 T_{II}: 33 T_{III}: 174.1 T_{solve}: 18.8

100 times speed up over building a fast direct solver from scratch for each θ

Replace a layer



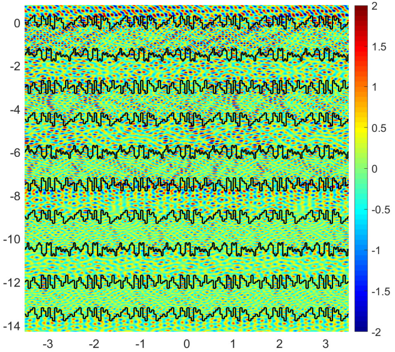
	Old	New Γ_4
N	121136	125184
T_I	2310	237
T_{II}	33.0	11.7
T_{III}	174.1	41.7
T_{solve}	18.8	15.7

Example from "A fast direct solution technique for quasi-periodic scattering in multi-layered medium," with Y. Zhang.

Change a wave number

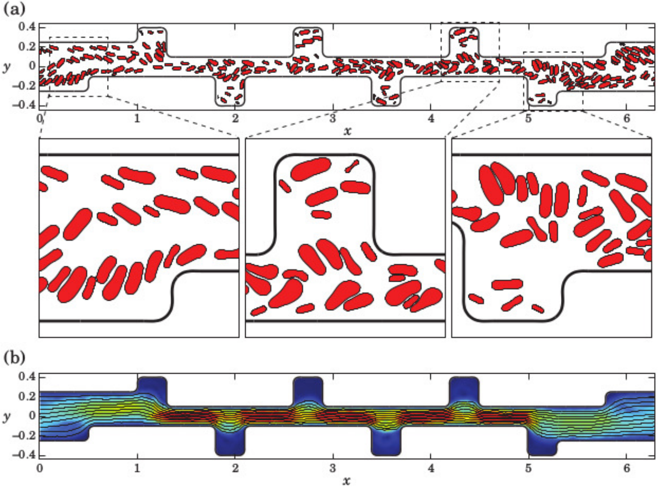
$$N = 121136$$

	$\omega_2 = 40\sqrt{2}$	$\omega_2 = 30$
T_I	2310	433
T_{II}	33.0	3.5
T_{III}	174.1	109.2
T_{solve}	18.8	13.6



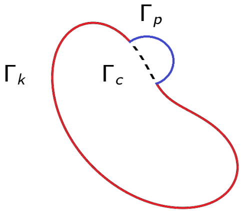
Example from "A fast direct solution technique for quasi-periodic scattering in multi-layered medium," with Y. Zhang.

Stokes flow



Example from "A fast algorithm for simulating multiphase flows through periodic geometries of arbitrary shape," with G. Marple, A. Barnett, and S. Veerapaneni.

The local perturbation problem



Consider the problem

$$\begin{aligned} -\Delta u(\mathbf{x}) &= 0, \qquad \mathbf{x} \in \Omega, \\ u(\mathbf{x}) &= g(\mathbf{x}), \quad \mathbf{x} \in \Gamma. \end{aligned}$$

We can express the discretized linear system as the following extended linear system

$$\begin{bmatrix} \mathbf{A}_{kk} & \mathbf{0} & \mathbf{A}_{kp} \\ \mathbf{A}_{ck} & \mathbf{A}_{cc} & \mathbf{0} \\ \mathbf{A}_{pk} & \mathbf{0} & \mathbf{A}_{pp} \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma}_k \\ \boldsymbol{\sigma}_c^{\text{dum}} \\ \boldsymbol{\sigma}_p \end{bmatrix} = \begin{bmatrix} \mathbf{g}_k \\ \mathbf{0} \\ \mathbf{g}_p \end{bmatrix}.$$

The extended linear system

This system can be expanded as

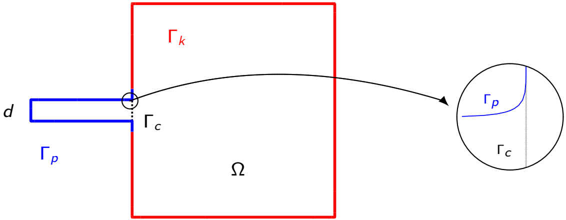
$$\left(\underbrace{\begin{bmatrix} \mathbf{A}_{oo} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{pp} \end{bmatrix}}_{\tilde{\mathbf{A}}} + \underbrace{\begin{bmatrix} \mathbf{0} & -\mathbf{A}_{kc} & \mathbf{A}_{kp} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{pk} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{Q}} \right) \underbrace{\begin{bmatrix} \boldsymbol{\sigma}_k \\ \boldsymbol{\sigma}_c^{\text{dum}} \\ \boldsymbol{\sigma}_p \end{bmatrix}}_{\boldsymbol{\sigma}_{\text{ext}}} = \underbrace{\begin{bmatrix} \mathbf{g}_k \\ \mathbf{0} \\ \mathbf{g}_p \end{bmatrix}}_{\mathbf{g}_{\text{ext}}}.$$

We can approximate the extended density via the following:

$$\begin{aligned} \boldsymbol{\sigma}_{\text{ext}} &= \left(\tilde{\mathbf{A}} + \mathbf{Q}\right)^{-1} \mathbf{g}_{\text{ext}} \\ &\approx \left(\tilde{\mathbf{A}} + \mathbf{LR}\right)^{-1} \mathbf{g}_{\text{ext}} \\ &\approx \tilde{\mathbf{A}}^{-1}\mathbf{g}_{\text{ext}} - \tilde{\mathbf{A}}^{-1}\mathbf{L}\left(\mathbf{I} + \mathbf{R}\tilde{\mathbf{A}}^{-1}\mathbf{L}\right)^{-1}\mathbf{R}\tilde{\mathbf{A}}^{-1}\mathbf{g}_{\text{ext}}, \end{aligned}$$

where $\mathbf{I}$ is an identity matrix, and $\mathbf{LR}$ denotes the low rank factorization of the update matrix $\mathbf{Q}$.

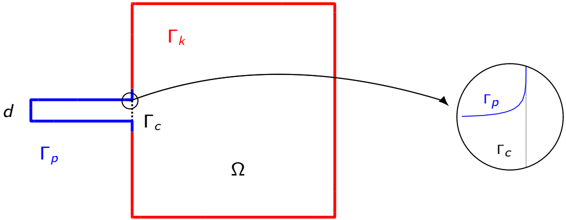
Numerical examples: Thinning nose



N_o	T_p	$T_{\mathrm{hbs},p}$	r_p	T_s	$T_{\mathrm{hbs},s}$	r_s
9232	4.83e-01	1.57e+00	3.25	1.12e-02	1.32e-02	1.18
18448	6.50e-01	2.38e+00	3.66	1.74e-02	1.46e-02	0.84
36880	1.11e+00	3.79e+00	3.42	4.00e-02	3.33e-02	0.83
73744	1.84e+00	6.38e+00	3.47	8.06e-02	7.04e-02	0.87
147472	3.56e+00	1.18e+01	3.33	1.71e-01	1.52e-01	0.89

Example from "An alternative extended linear system for boundary value problems on locally perturbed geometries," with Y. Zhang.

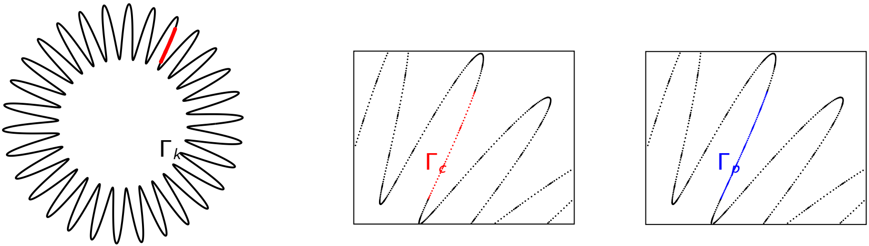
Numerical examples: Fixed nose



N_o	N_c	T_p	$T_{\text{hbs},p}$	r_p	T_s	$T_{\text{hbs},s}$	r_s
9344	128	5.10e-01	1.28e+00	2.50	1.02e-02	7.92e-03	0.77
18688	256	9.25e-01	2.18e+00	2.36	2.15e-02	1.59e-02	0.74
37376	512	1.30e+00	3.49e+00	2.69	3.97e-02	3.00e-02	0.76
74752	1024	2.31e+00	6.63e+00	2.87	8.67e-02	6.40e-02	0.74
149504	2048	4.06e+00	1.19e+01	2.92	1.71e-01	1.61e-01	0.94

Example from "An alternative extended linear system for boundary value problems on locally perturbed geometries," with Y. Zhang.

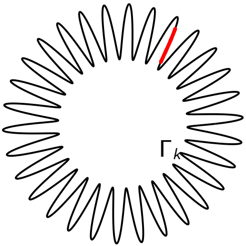
Local refinement: Laplace

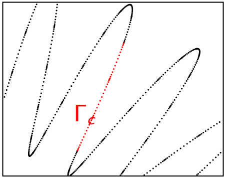


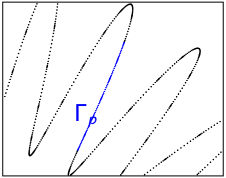
N_p	$\frac{N_p}{N_o}$	T_p	$T_{\mathrm{hbs},p}$	r_p	T_s	$T_{\mathrm{hbs},s}$	r_s
96	0.015	5.03e-01	7.52e+00	14.9	1.30e-02	1.32e-02	1.02
192	0.03	3.62e-01	7.77e+00	21.4	1.25e-02	9.30e-03	0.74
384	0.06	3.90e-01	7.72e+00	19.8	1.42e-02	9.13e-03	0.64
768	0.12	4.11e-01	7.78e+00	18.9	1.20e-02	9.06e-03	0.76
1536	0.24	6.09e-01	8.03e+00	13.2	1.66e-02	1.00e-02	0.60

Example from "An alternative extended linear system for boundary value problems on locally perturbed geometries," with Y. Zhang.

Local refinement: Helmholtz



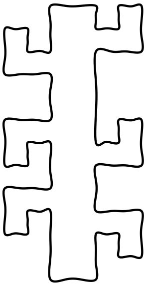


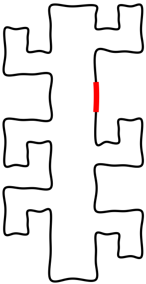


N_p	$\frac{N_p}{N_o}$	T_p	$T_{\mathrm{hbs},p}$	r_p	T_s	$T_{\mathrm{hbs},s}$	r_s
96	0.015	1.13e+00	3.97e+01	35.2	4.06e-02	2.86e-02	0.71
192	0.03	1.36e+00	4.08e+01	29.9	4.64e-02	2.93e-02	0.63
384	0.06	1.44e+00	4.08e+01	28.4	3.91e-02	2.54e-02	0.65
768	0.12	1.64e+00	4.17e+01	25.4	3.69e-02	2.70e-02	0.73
1536	0.24	2.64e+00	4.08e+01	15.4	4.39e-02	3.33e-02	0.76

Example from "An alternative extended linear system for boundary value problems on locally perturbed geometries," with Y.Zhang.

Stokes: Local refinement

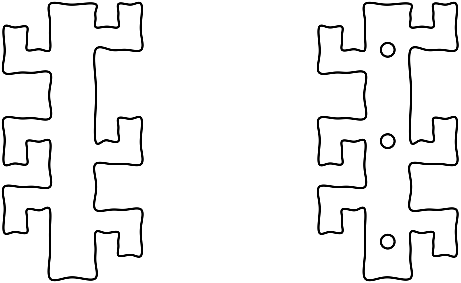




N_{channel}, N_c, N_p	T_{precomp}	T_{sol}	E	$\tilde{T}_{\mathrm{precomp}}$	$\tilde{T}_{\mathrm{sol}}$	$\tilde{E}$
1920, 16, 64	21.76	0.033	9.49e-9	0.33	0.008	8.47e-9
3840, 32, 128	36.07	0.037	1.22e-9	0.64	0.020	1.08e-9
7680, 64, 256	48.22	0.048	1.78e-10	1.12	0.024	3.67e-10
15360, 96, 384	69.72	0.077	2.40e-10	1.69	0.041	3.05e-10

Example from "A fast direct solver for integral equations on locally refined boundary discretizations and its application to multiphase flow simulations," with Y. Zhang and S. Veerapaneni.

Stokes: Objects inside



N_{channel}	$\tilde{T}_{\text{precomp}}$	$\tilde{T}_{\text{sol}}$	$\tilde{E}$
1920	2.84	0.010	4.02e-9
3840	4.65	0.018	4.99e-10
7680	8.50	0.030	1.26e-10
15360	15.78	0.049	4.57e-11

Example from "A fast direct solver for integral equations on locally refined boundary discretizations and its application to multiphase flow simulations," with Y. Zhang and S. Veerapaneni.

Concluding remarks

Summary

- A brief introduction to integral equations.
- Linear scaling methods.
  1D boundary integral equations, systems arising from the discretization of non-oscillatory PDEs.
- Great for problems with multiple right-hand sides.
  A solver for periodic scattering is 600 times faster than existing techniques for a problem with 200 right-hand sides.
- Extensions are increasing the range of applicability of boundary integral equations: Periodic boundary value problems, locally perturbed geometries, adaptive discretizations with fast solvers for periodic problems. In applications these techniques can result in hundreds to thousands of times speed up.