Fast direct solvers for elliptic partial differential equations on locally-perturbed geometries

Yabin Zhang

RICE

(Joint work with **Adrianna Gillman**)

Definition of “fast”

A numerical linear algebraic method is **fast** if its execution time scales asymptotically less than the cost of classic linear algebra techniques.

Definition of "fast"

A numerical linear algebraic method is **fast** if its execution time scales asymptotically less than the cost of classic linear algebra techniques.

What is a "direct solver"?

Given a pre-set tolerance ϵ and a linear system $\boldsymbol{A}x = b$, a **direct solver** constructs an operator $\boldsymbol{T}$ so that $\|\boldsymbol{A}^{-1} - \boldsymbol{T}\| \leq \epsilon$.

Definition of “fast”

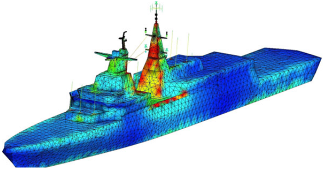
A numerical linear algebraic method is **fast** if its execution time scales asymptotically less than the cost of classic linear algebra techniques.

What is a “direct solver”?

Given a pre-set tolerance ϵ and a linear system $\boldsymbol{A}x = b$, a **direct solver** constructs an operator $\boldsymbol{T}$ so that $\|\boldsymbol{A}^{-1} - \boldsymbol{T}\| \leq \epsilon$.

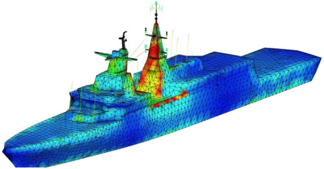
For a direct solver to be fast, the cost of constructing $\boldsymbol{T}$ and applying $\boldsymbol{T}$ to a vector needs to be low.

Motivation

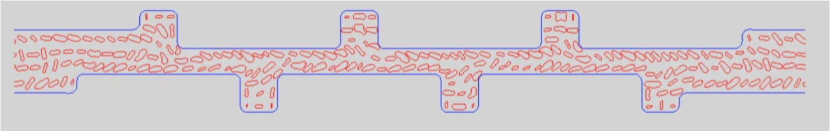


https://altairhyperworks.com/product/FEKO/Applications-Antenna-Placement

Motivation



https://altairhyperworks.com/product/FEKO/Applications-Antenna-Placement

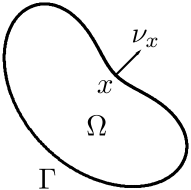


G Marple, A. Barnett, A. Gillman, and A. Veerapaneni, *A Fast Algorithm for Simulating Multiphase Flows Through Periodic Geometries of Arbitrary Shape.*

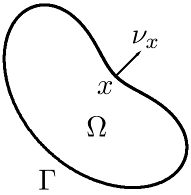
Model problem

Consider the Laplace BVP

$$
\begin{aligned}
-\Delta u(x) &= 0 && \text{for } x \in \Omega, \\
u(x) &= f(x) && \text{for } x \in \Gamma = \partial\Omega.
\end{aligned}
$$



Model problem



Consider the Laplace BVP

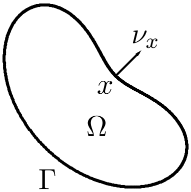
$$\begin{aligned} -\Delta u(x) &= 0 && \text{for } x \in \Omega, \\ u(x) &= f(x) && \text{for } x \in \Gamma = \partial\Omega. \end{aligned}$$

The solution to the BVP can be represented as a double-layer potential

$$u(x) = \int_\Gamma \frac{\partial G(x,y)}{\partial \nu_y} \sigma(y) dl(y),\ x \in \Omega$$

where $\sigma(x)$ is an unknown boundary charge density and $G(x,y) = -\frac{1}{2\pi} \log\left(\frac{1}{|x-y|}\right)$ is the Green's function.

Model problem



Consider the Laplace BVP

$$\begin{aligned} -\Delta u(x) &= 0 && \text{for } x \in \Omega, \\ u(x) &= f(x) && \text{for } x \in \Gamma = \partial\Omega. \end{aligned}$$

The solution to the BVP can be represented as a double-layer potential

$$u(x) = \int_\Gamma \frac{\partial G(x,y)}{\partial \nu_y}\sigma(y)dl(y), \ x \in \Omega$$

where $\sigma(x)$ is an unknown boundary charge density and $G(x,y) = -\frac{1}{2\pi}\log\left(\frac{1}{|x-y|}\right)$ is the Green's function.

Enforcing the boundary condition yields the **boundary integral equation** (BIE)

$$-\frac{1}{2}\sigma(x) + \int_\Gamma \frac{\partial G(x,y)}{\partial \nu_y}\sigma(y)dl(y) = f(x), \text{ for } x \in \Gamma.$$

The discretized linear system

Let $\vec{\sigma} = (\sigma(x_1), \ldots, \sigma(x_n))^T$, $\vec{f} = (f(x_1), \ldots, f(x_n))^T$, $\boldsymbol{I}$ be the identity matrix, and $\boldsymbol{D}$ be a matrix with entries $\boldsymbol{D}_{ij} = \frac{\partial G(x_i, x_j)}{\partial \nu_{x_j}} w_j$, then the discretized BIE can be written as

$$\boldsymbol{A}\vec{\sigma} = (-\frac{1}{2}\boldsymbol{I} + \boldsymbol{D})\vec{\sigma} = \vec{f}$$

The discretized linear system

Let $\vec{\sigma} = (\sigma(x_1), \ldots, \sigma(x_n))^T$, $\vec{f} = (f(x_1), \ldots, f(x_n))^T$, $\boldsymbol{I}$ be the identity matrix, and $\boldsymbol{D}$ be a matrix with entries $\boldsymbol{D}_{ij} = \frac{\partial G(x_i, x_j)}{\partial \nu_{x_j}} w_j$, then the discretized BIE can be written as

$$\boldsymbol{A}\vec{\sigma} = (-\frac{1}{2}\boldsymbol{I} + \boldsymbol{D})\vec{\sigma} = \vec{f}$$

$\boldsymbol{A}$ is called the *coefficient matrix*.

The discretized linear system

Let $\vec{\sigma} = (\sigma(x_1), \ldots, \sigma(x_n))^T$, $\vec{f} = (f(x_1), \ldots, f(x_n))^T$, $\boldsymbol{I}$ be the identity matrix, and $\boldsymbol{D}$ be a matrix with entries $\boldsymbol{D}_{ij} = \frac{\partial G(x_i,x_j)}{\partial \nu_{x_j}} w_j$, then the discretized BIE can be written as

$$\boldsymbol{A}\vec{\sigma} = (-\frac{1}{2}\boldsymbol{I} + \boldsymbol{D})\vec{\sigma} = \vec{f}$$

$\boldsymbol{A}$ is called the *coefficient matrix*.

Properties of the coefficient matrix $\boldsymbol{A}$:

- $\boldsymbol{A}$ is a dense matrix.
- The size of $\boldsymbol{A}$ depends on the number of discretization points N on the boundary Γ.
- $\boldsymbol{A}$ is data-sparse.
    - Particularly, the off-diagonal blocks of $\boldsymbol{A}$ are low-rank.

Data-sparse property of the coefficient matrix

Definition: A matrix $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ is ϵ-rank if it has exactly $k = k(\epsilon)$ singular values that are greater than ϵ. $\boldsymbol{S}$ is called a low-rank matrix if $k << m$.

Data-sparse property of the coefficient matrix

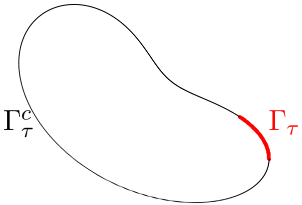
Definition: A matrix $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ is ϵ-rank if it has exactly $k = k(\epsilon)$ singular values that are greater than ϵ. $\boldsymbol{S}$ is called a low-rank matrix if $k << m$.

Let's verify that the off-diagonal blocks of the coefficient matrix $\boldsymbol{A}$ are indeed low-rank by an example:

Data-sparse property of the coefficient matrix

Definition: A matrix $\boldsymbol{S} \in \mathbb{R}^{m \times n}$ is ϵ-rank if it has exactly $k = k(\epsilon)$ singular values that are greater than ϵ. $\boldsymbol{S}$ is called a low-rank matrix if $k << m$.

Let's verify that the off-diagonal blocks of the coefficient matrix $\boldsymbol{A}$ are indeed low-rank by an example:



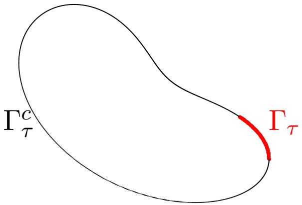
Boundary: $\Gamma = \Gamma_\tau \cup \Gamma_\tau^c$

Matrix block: $\boldsymbol{A}(\Gamma_\tau, \Gamma_\tau^c) \in \mathbb{R}^{100 \times 900}$

Data-sparse property of the coefficient matrix

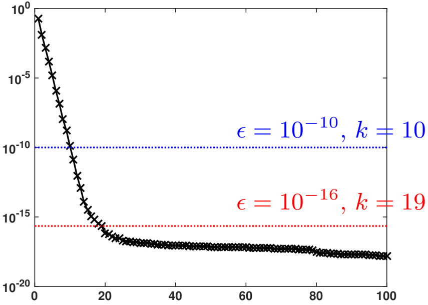
Definition: A matrix $\boldsymbol{S} \in \mathbb{R}^{m\times n}$ is ϵ-rank if it has exactly $k = k(\epsilon)$ singular values that are greater than ϵ. $\boldsymbol{S}$ is called a low-rank matrix if $k << m$.

Let's verify that the off-diagonal blocks of the coefficient matrix $\boldsymbol{A}$ are indeed low-rank by an example:



Boundary: $\Gamma = \Gamma_\tau \cup \Gamma_\tau^c$

Matrix block: $\boldsymbol{A}(\Gamma_\tau, \Gamma_\tau^c) \in \mathbb{R}^{100\times 900}$



The singular values of $\boldsymbol{A}(\Gamma_\tau, \Gamma_\tau^c)$

Block-separable matrix

A matrix $\boldsymbol{A}$ of dimension $(np) \times (np)$ is *block-separable* if it consists $p \times p$ blocks each of size $n \times n$: e.g. for $p = 4$,

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{D}_{11} & \boldsymbol{A}_{12} & \boldsymbol{A}_{13} & \boldsymbol{A}_{14} \\ \boldsymbol{A}_{21} & \boldsymbol{D}_{22} & \boldsymbol{A}_{23} & \boldsymbol{A}_{24} \\ \boldsymbol{A}_{31} & \boldsymbol{A}_{32} & \boldsymbol{D}_{33} & \boldsymbol{A}_{34} \\ \boldsymbol{A}_{41} & \boldsymbol{A}_{42} & \boldsymbol{A}_{43} & \boldsymbol{D}_{44} \end{bmatrix}.$$

And each of the off-diagonal block admits the factorization

$$\begin{array}{ccccc} \boldsymbol{A}_{ij} & = & \boldsymbol{U}_i & \tilde{\boldsymbol{A}}_{ij} & \boldsymbol{V}_j^* \\ n \times n & & n \times k & k \times k & k \times n \end{array}$$

where the rank k is significantly smaller than the block size n.

A. Gillman, P. Young, and P.G. Martinsson, *A direct solver with O(N) complexity for integral equations on one-dimensional domains*

Then we have $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{D}_{11} & \boldsymbol{U}_1\tilde{\boldsymbol{A}}_{12}\boldsymbol{V}_2^* & \boldsymbol{U}_1\tilde{\boldsymbol{A}}_{13}\boldsymbol{V}_3^* & \boldsymbol{U}_1\tilde{\boldsymbol{A}}_{14}\boldsymbol{V}_4^* \\ \boldsymbol{U}_2\tilde{\boldsymbol{A}}_{21}\boldsymbol{V}_1^* & \boldsymbol{D}_{22} & \boldsymbol{U}_2\tilde{\boldsymbol{A}}_{23}\boldsymbol{V}_3^* & \boldsymbol{U}_2\tilde{\boldsymbol{A}}_{24}\boldsymbol{V}_4^* \\ \boldsymbol{U}_3\tilde{\boldsymbol{A}}_{31}\boldsymbol{V}_1^* & \boldsymbol{U}_3\tilde{\boldsymbol{A}}_{32}\boldsymbol{V}_2^* & \boldsymbol{D}_{33} & \boldsymbol{U}_3\tilde{\boldsymbol{A}}_{34}\boldsymbol{V}_4^* \\ \boldsymbol{U}_4\tilde{\boldsymbol{A}}_{41}\boldsymbol{V}_1^* & \boldsymbol{U}_4\tilde{\boldsymbol{A}}_{42}\boldsymbol{V}_2^* & \boldsymbol{U}_4\tilde{\boldsymbol{A}}_{43}\boldsymbol{V}_3^* & \boldsymbol{D}_{44} \end{bmatrix}$,

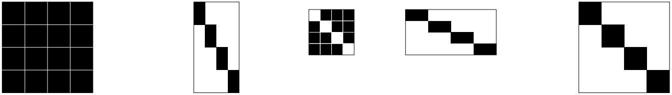
and it can be factored as

$$\boldsymbol{A} = \underbrace{\begin{bmatrix} \boldsymbol{U}_1 & & & \\ & \boldsymbol{U}_2 & & \\ & & \boldsymbol{U}_3 & \\ & & & \boldsymbol{U}_4 \end{bmatrix}}_{=\boldsymbol{U}} \underbrace{\begin{bmatrix} \boldsymbol{0} & \tilde{\boldsymbol{A}}_{12} & \tilde{\boldsymbol{A}}_{13} & \tilde{\boldsymbol{A}}_{14} \\ \tilde{\boldsymbol{A}}_{21} & \boldsymbol{0} & \tilde{\boldsymbol{A}}_{23} & \tilde{\boldsymbol{A}}_{24} \\ \tilde{\boldsymbol{A}}_{31} & \tilde{\boldsymbol{A}}_{32} & \boldsymbol{0} & \tilde{\boldsymbol{A}}_{34} \\ \tilde{\boldsymbol{A}}_{41} & \tilde{\boldsymbol{A}}_{42} & \tilde{\boldsymbol{A}}_{43} & \boldsymbol{0} \end{bmatrix}}_{=\tilde{\boldsymbol{A}}} \underbrace{\begin{bmatrix} \boldsymbol{V}_1^* & & & \\ & \boldsymbol{V}_2^* & & \\ & & \boldsymbol{V}_3^* & \\ & & & \boldsymbol{V}_4^* \end{bmatrix}}_{=\boldsymbol{V}^*} + \underbrace{\begin{bmatrix} \boldsymbol{D}_{11} & & & \\ & \boldsymbol{D}_{22} & & \\ & & \boldsymbol{D}_{33} & \\ & & & \boldsymbol{D}_{44} \end{bmatrix}}_{=\boldsymbol{D}},$$

Block separable matrix and its inversion

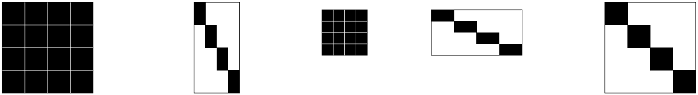
$\boldsymbol{A}$ admits the factorization:

$$\underset{pn \times pn}{\boldsymbol{A}} = \underset{pn \times pk}{\boldsymbol{U}} \quad \underset{pk \times pk}{\tilde{\boldsymbol{A}}} \quad \underset{pk \times pn}{\boldsymbol{V}^*} + \underset{pn \times pn}{\boldsymbol{D}},$$



Lemma (Variation of Woodbury) If $\boldsymbol{A}$ admits the factorization above, the inverse can be evaluated as

$$\underset{pn \times pn}{\boldsymbol{A}^{-1}} = \underset{pn \times pk}{\boldsymbol{E}} \quad \underset{pk \times pk}{(\tilde{\boldsymbol{A}} + \hat{\boldsymbol{D}})^{-1}} \quad \underset{pk \times pn}{\boldsymbol{F}^*} + \underset{pn \times pn}{\boldsymbol{G}},$$



where (provided all intermediate matrices are invertible)
$\hat{\boldsymbol{D}} = \left(\boldsymbol{V}^* \boldsymbol{D}^{-1} \boldsymbol{U}\right)^{-1}$, $\boldsymbol{E} = \boldsymbol{D}^{-1} \boldsymbol{U} \hat{\boldsymbol{D}}$, $\boldsymbol{F} = (\hat{\boldsymbol{D}} \boldsymbol{V}^* \boldsymbol{D}^{-1})^*$, and
$\boldsymbol{G} = \boldsymbol{D}^{-1} - \boldsymbol{D}^{-1} \boldsymbol{U} \hat{\boldsymbol{D}} \boldsymbol{V}^* \boldsymbol{D}^{-1}$.

Hierarchically block separable(HBS) matrix

The lemma reduces the cost of inversion from $(pn)^3$ to $(pk)^3$!

Hierarchically block separable(HBS) matrix

The lemma reduces the cost of inversion from $(pn)^3$ to $(pk)^3$!

But this is not "fast" yet.

We obtain a fast scheme by performing the above factorization "hierarchically".

Hierarchically block separable(HBS) matrix

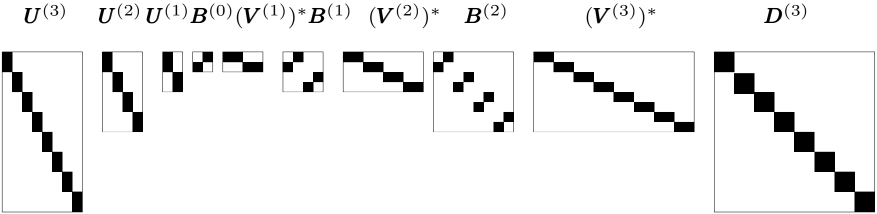
The lemma reduces the cost of inversion from $(pn)^3$ to $(pk)^3$!

But this is not "fast" yet.

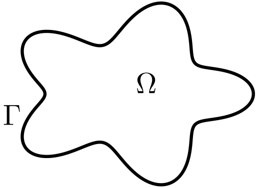
We obtain a fast scheme by performing the above factorization "hierarchically".
For example, a "3-level" telescoping factorization of $\boldsymbol{A}$ will be

$$\boldsymbol{A} = \boldsymbol{U}^{(3)}\big(\boldsymbol{U}^{(2)}\big(\boldsymbol{U}^{(1)}\,\boldsymbol{B}^{(0)}\,(\boldsymbol{V}^{(1)})^*\big) + \boldsymbol{B}^{(1)}\big)(\boldsymbol{V}^{(2)})^* + \boldsymbol{B}^{(2)}\big)(\boldsymbol{V}^{(3)})^* + \boldsymbol{D}^{(3)}.$$

And the block structure will look like:



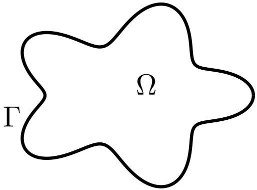
Numerical examples



Consider the BIE

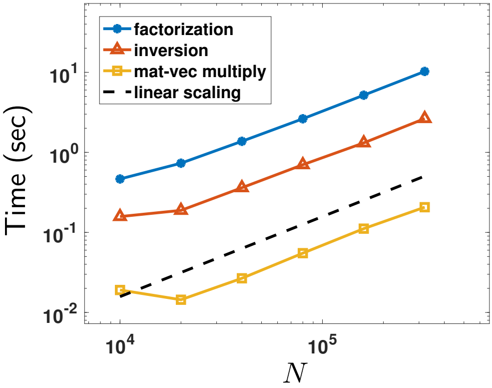
$$-\frac{1}{2}\sigma(x)+\int_\Gamma \frac{\partial G(x,y)}{\partial \nu_y}\sigma(y)dl(y) = f(x), \text{ for } x \in \Gamma.$$

Numerical examples



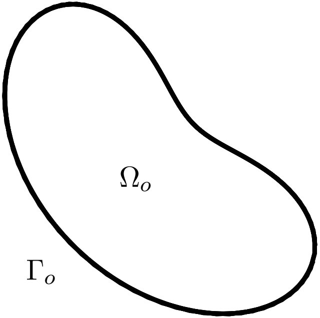
Consider the BIE

$$-\frac{1}{2}\sigma(x)+\int_\Gamma \frac{\partial G(x,y)}{\partial \nu_y}\sigma(y)dl(y) = f(x),\ \text{for } x \in \Gamma.$$



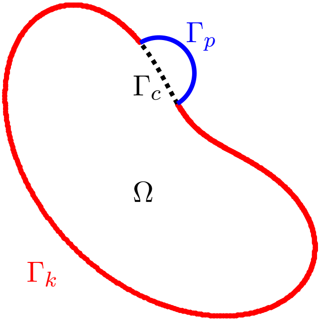
Problem with locally perturbed geometry

Consider a BIE defined on Γ_o.
We can solve this by building a direct solver.

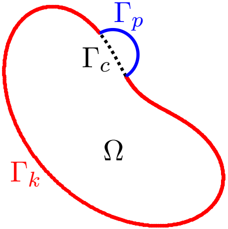


Problem with locally perturbed geometry

Now, suppose we already have a direct solver for $\Gamma_o = \Gamma_k \cup \Gamma_c$.
We want to solve the BIE defined on $\Gamma := \Gamma_k \cup \Gamma_p$

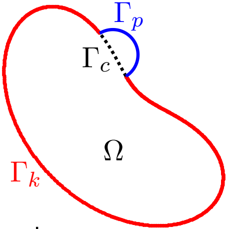


Problem with locally perturbed geometry



We have a direct solver for $\Gamma_o = \Gamma_k \cup \Gamma_c$.
We want to solve a BIE defined on $\Gamma = \Gamma_k \cup \Gamma_p$

Problem with locally perturbed geometry



We have a direct solver for $\Gamma_o = \Gamma_k \cup \Gamma_c$.
We want to solve a BIE defined on $\Gamma = \Gamma_k \cup \Gamma_p$

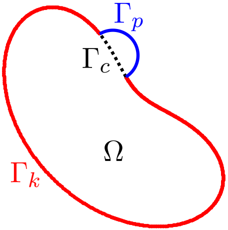
The discretized integral equation on Γ can be expressed as

$$\left(\underbrace{\begin{bmatrix} \boldsymbol{A}_{oo} & 0 \\ 0 & \boldsymbol{A}_{pp} \end{bmatrix}}_{\boldsymbol{A}} + \underbrace{\begin{bmatrix} 0 & \begin{pmatrix} -\boldsymbol{A}_{kc} \\ -\boldsymbol{B}_{cc} \end{pmatrix} & \boldsymbol{A}_{op} \\ \boldsymbol{A}_{pk} & 0 & 0 \end{bmatrix}}_{\boldsymbol{M}} \right) \begin{pmatrix} \boldsymbol{\sigma}_k \\ \boldsymbol{\sigma}_c \\ \boldsymbol{\sigma}_p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_k \\ 0 \\ \boldsymbol{f}_p \end{pmatrix}$$

where $\boldsymbol{B}_{cc}$ equals to $\boldsymbol{A}_{cc}$ with diagonal entries set to zero, $\boldsymbol{A}_{oo}$ denotes the interaction matrix on Γ_o, $\boldsymbol{A}_{kc}$ denotes the interaction between Γ_k and Γ_c, and the rest follows the same notation.

L. Greengard, D. Gueyffier, P.G. Martinsson, V. Rokhlin, *Fast direct solvers for integral equations in complex three-dimensional domains*

A closer look at the update matrix $\boldsymbol{M}$

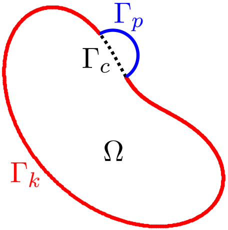


$\boldsymbol{M}$ has three low-rank sub-blocks:

$\boldsymbol{A}_{pk} \approx \boldsymbol{L}_{pk}\boldsymbol{R}_{pk}, \quad \boldsymbol{A}_{kc} \approx \boldsymbol{L}_{kc}\boldsymbol{R}_{kc},$

and $\boldsymbol{A}_{op} \approx \boldsymbol{L}_{op}\boldsymbol{R}_{op}$.

A closer look at the update matrix $\boldsymbol{M}$



$\boldsymbol{M}$ has three low-rank sub-blocks:

$\boldsymbol{A}_{pk} \approx \boldsymbol{L}_{pk}\boldsymbol{R}_{pk}, \quad \boldsymbol{A}_{kc} \approx \boldsymbol{L}_{kc}\boldsymbol{R}_{kc},$

and $\boldsymbol{A}_{op} \approx \boldsymbol{L}_{op}\boldsymbol{R}_{op}$.

Combining the three factorizations, we obtain a low-rank factorization of the update matrix:

$$\boldsymbol{M} \approx \underbrace{\begin{bmatrix} 0 & \begin{pmatrix} -\boldsymbol{L}_{kc} & 0 \\ 0 & -\boldsymbol{B}_{cc} \end{pmatrix} & \boldsymbol{L}_{op} \\ \boldsymbol{L}_{pk} & 0 & 0 \end{bmatrix}}_{\boldsymbol{L}} \underbrace{\begin{bmatrix} \boldsymbol{R}_{pk} & 0 & 0 \\ 0 & \begin{pmatrix} \boldsymbol{R}_{kc} \\ \boldsymbol{I} \end{pmatrix} & 0 \\ 0 & 0 & \boldsymbol{R}_{op} \end{bmatrix}}_{\boldsymbol{R}}$$

Why building a low-rank factorization of $\boldsymbol{M}$?

The inverse of $(\boldsymbol{A}+\boldsymbol{M})$ can be approximated as

$$\underbrace{(\boldsymbol{A}+\boldsymbol{L}\boldsymbol{R})^{-1}}_{N\times N} \quad = \quad \boldsymbol{A}^{-1} \quad + \quad \boldsymbol{A}^{-1}\boldsymbol{L} \quad \underbrace{(\boldsymbol{I}+\boldsymbol{R}\boldsymbol{A}^{-1}\boldsymbol{L})^{-1}}_{K\times K} \quad \boldsymbol{R}\boldsymbol{A}^{-1}$$

Why building a low-rank factorization of $\boldsymbol{M}$?

The inverse of $(\boldsymbol{A}+\boldsymbol{M})$ can be approximated as

$$\underbrace{(\boldsymbol{A}+\boldsymbol{L}\boldsymbol{R})^{-1}}_{N\times N} = \boldsymbol{A}^{-1} + \boldsymbol{A}^{-1}\boldsymbol{L} \underbrace{(\boldsymbol{I}+\boldsymbol{R}\boldsymbol{A}^{-1}\boldsymbol{L})^{-1}}_{K\times K} \boldsymbol{R}\boldsymbol{A}^{-1}$$

The solution to the extended system can be approximated as

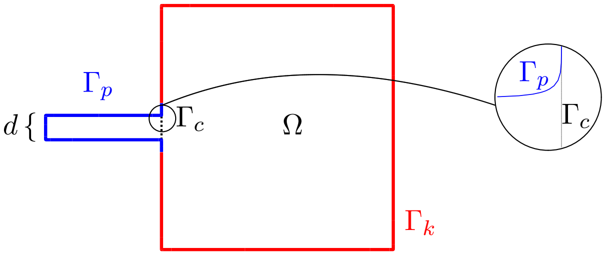
$$(\boldsymbol{A}+\boldsymbol{M})^{-1}\boldsymbol{f} \approx \boldsymbol{A}^{-1}\boldsymbol{f} + \boldsymbol{A}^{-1}\boldsymbol{L}(\boldsymbol{I}+\boldsymbol{R}\boldsymbol{A}^{-1}\boldsymbol{L})^{-1}\boldsymbol{R}\boldsymbol{A}^{-1}\boldsymbol{f}.$$

The existing direct solver for the BIE on Γ_o can be reused to calculate the repeated terms

$$\boldsymbol{A}^{-1}\boldsymbol{f} = \begin{bmatrix} \boldsymbol{A}_{oo}^{-1} & 0 \\ 0 & \boldsymbol{A}_{pp}^{-1} \end{bmatrix} \begin{pmatrix} \boldsymbol{f}_k \\ 0 \\ \boldsymbol{f}_p \end{pmatrix} \text{ and } \boldsymbol{A}^{-1}\boldsymbol{L} = \begin{bmatrix} \boldsymbol{A}_{oo}^{-1} & 0 \\ 0 & \boldsymbol{A}_{pp}^{-1} \end{bmatrix} \boldsymbol{L}.$$

Numerical tests

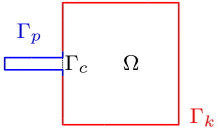
Consider the Laplace BVP defined on the "square with thinning nose geometry":

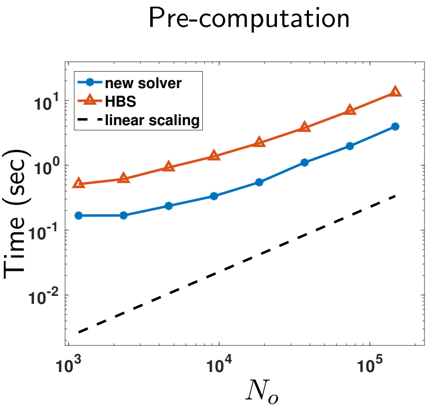


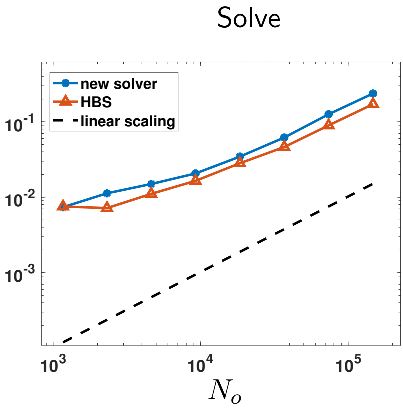
- d decreases as N_o increases so that $N_c = 16$ remains a constant.

Corners are smoothed by the method in C. Eptein and M. O'Neil, *Smoothed corners and scattered waves*.

Laplace on a square with thinning nose







($N_c = 16$, and $N_p \in [700, 900]$.)

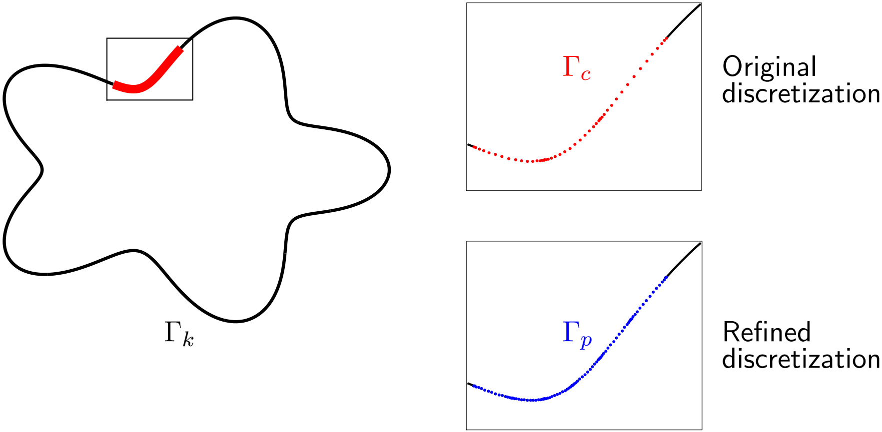
Laplace on a square with thinning nose

N_o	$T_{new,p}$	$T_{hbs,p}$	$\frac{T_{new,p}}{T_{hbs,p}}$	$T_{new,s}$	$T_{hbs,s}$	$\frac{T_{new,s}}{T_{hbs,s}}$
4624	0.24	0.92	0.26	1.5e-02	1.1e-02	1.4
9232	0.33	1.37	0.24	2.0e-02	1.6e-02	1.3
18448	0.55	2.20	0.25	3.5e-02	2.8e-02	1.2
36880	1.10	3.76	0.29	6.2e-02	4.6e-02	1.3
73744	1.98	6.88	0.29	0.13	9.0e-02	1.4
147472	4.00	13.2	0.30	0.24	0.17	1.4

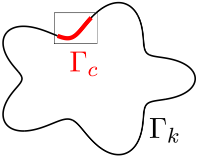
- With $\epsilon = 1 \times 10^{-10}$, the relative error is around 1×10^{-9}.
- New solver scales linearly w.r.t. N_o.
- In terms of total cost, it would take 100 to 260 solves to make the new solver slower than building a new HBS solver from scratch.

Numerical tests

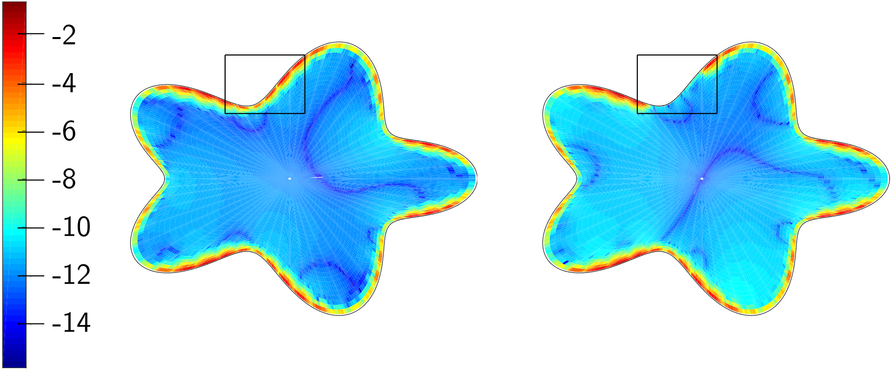
Consider the Laplace BVP defined on the smooth star with the boxed segment locally refined:



Star with locally refined discretization

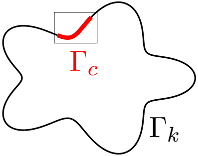


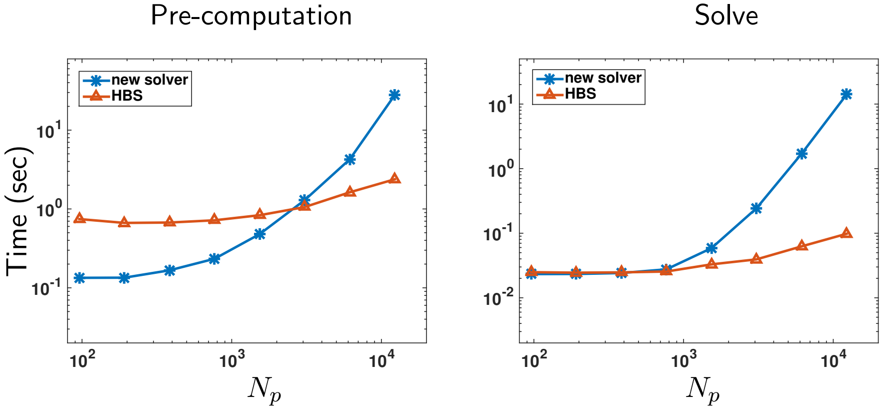
Relative error on a log10 scale:



Original discretization Refined discretization

Star with locally refined discretization





$(N_k = 592, N_c = 48$ remain constant.)

Star with locally refined discretization

N_p	$T_{new,p}$	$T_{hbs,p}$	$\frac{T_{new,p}}{T_{hbs,p}}$	$T_{new,s}$	$T_{hbs,s}$	$\frac{T_{new,s}}{T_{hbs,s}}$
96	4.2e-02	0.20	0.21	4.3e-03	5.7e-03	0.75
192	4.9e-02	0.191	0.25	3.5e-03	3.5e-03	1.00
384	7.0e-02	0.20	0.34	4.5e-03	4.1e-03	1.11
768	0.13	0.24	0.55	8.3e-03	5.4e-03	1.54
1536	0.34	0.32	1.07	3.5e-02	9.8e-03	3.60

- The new solver can be incorporated into an adaptive discretization technique for BIEs if the local refinement only adds a reasonable number of new points.
- For N_p large, the new solver is much more expensive than HBS. Cost is dominated by $\boldsymbol{A}_{pp}^{-1}$.

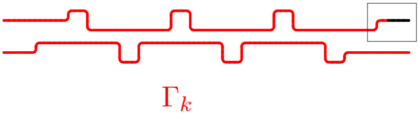
Application in modeling objects in Stokes flow

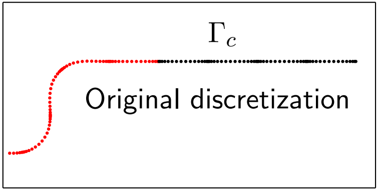
(click for video)

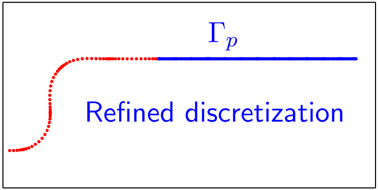
Example is from G. Marple, A. Barnett, A. Gillman, and S. Veerapaneni, *A fast algorithm for simulating multiphase flows through periodic geometries of arbitrary shape.*

Stokes on locally refined periodic pipes

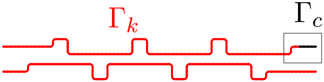
Consider the periodic Stokes problem defined on the following pipe geometry. (The boundary wall consists infinite copies of the shown piece.)

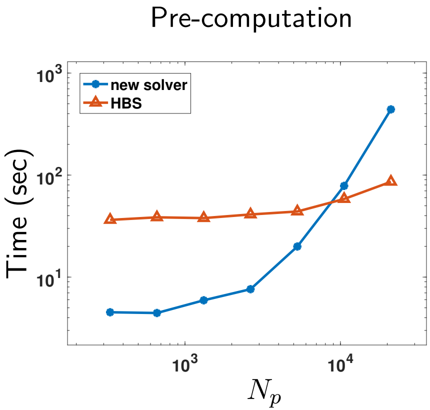


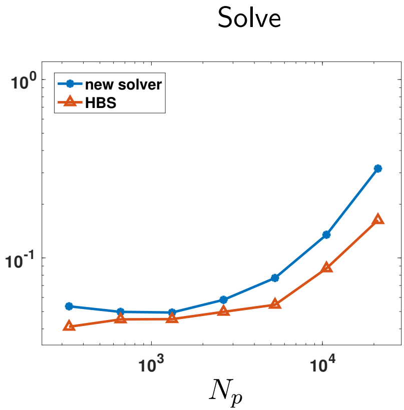




Stokes on locally refined periodic pipes







($N_k = 6290$ and $Nc = 110$ remain constant.)

Stokes on locally refined periodic pipes

N_p	$T_{new,p}$	$T_{hbs,p}$	$\frac{T_{new,p}}{T_{hbs,p}}$	$T_{new,s}$	$T_{hbs,s}$	$\frac{T_{new,s}}{T_{hbs,s}}$
330	4.5e+00	3.6e+01	0.12	5.4e-02	4.1e-02	1.3
660	4.4e+00	3.9e+01	0.12	5.0e-02	4.5e-02	1.1
1320	5.9e+00	3.8e+01	0.16	4.9e-02	4.5e-02	1.1
2640	7.6e+00	4.1e+01	0.19	5.8e-02	5.0e-02	1.2
5280	2.0e+01	4.4e+01	0.45	7.7e-02	5.5e-02	1.4

- With tolerance for (matrix) low-rank approximation $\epsilon = 1 \times 10^{-12}$, the relative error is about 3×10^{-8}.

Conclusion

Summary

- A brief introduction to fast direct solvers for BIEs and particularly the Hierarchically block-sparable (HBS) solver.
  - Linear scaling 2D
  - Great for problems with multipole right-hand-sides
- A new fast direct solver for problems defined on locally-perturbed geometries
  - Reuses the inverse approximation previously constructed for the original geometry
  - Outperforms HBS from scratch when the size of changes is small.
  - Very efficient in handling local refinement in discretization

Conclusion

Summary

- A brief introduction to fast direct solvers for BIEs and particularly the Hierarchically block-sparable (HBS) solver.
  - Linear scaling 2D
  - Great for problems with multipole right-hand-sides
- A new fast direct solver for problems defined on locally-perturbed geometries
  - Reuses the inverse approximation previously constructed for the original geometry
  - Outperforms HBS from scratch when the size of changes is small.
  - Very efficient in handling local refinement in discretization

Conclusion

Summary

- A brief introduction to fast direct solvers for BIEs and particularly the Hierarchically block-sparable (HBS) solver.
  - Linear scaling 2D
  - Great for problems with multipole right-hand-sides
- A new fast direct solver for problems defined on locally-perturbed geometries
  - Reuses the inverse approximation previously constructed for the original geometry
  - Outperforms HBS from scratch when the size of changes is small.
  - Very efficient in handling local refinement in discretization

Conclusion

Summary

- A brief introduction to fast direct solvers for BIEs and particularly the Hierarchically block-sparable (HBS) solver.
    - Linear scaling 2D
    - Great for problems with multipole right-hand-sides
- A new fast direct solver for problems defined on locally-perturbed geometries
    - Reuses the inverse approximation previously constructed for the original geometry
    - Outperforms HBS from scratch when the size of changes is small.
    - Very efficient in handling local refinement in discretization

Future directions

- Continue on building an adaptive discretization technique for Stokes and a fast direct solver that works with it.
- 3D problems.