Fast direct solvers for integral equations

P.G. Martinsson

Yale University

Joint work with Vladimir Rokhlin and Mark Tygert.

We consider the integral equation

$$u(x) + \int_\Gamma K(x,y)u(y)\,ds(y) = f(x), \qquad x \in \Gamma. \tag{1}$$

Upon discretization, equation (1) turns into a discrete equation

$$(I + A)u = f \tag{2}$$

where A is a (typically dense) $N \times N$ matrix.

Equation (2) can be solved rapidly using an ***iterative method:***

$$\text{Cost} = n_{\text{iter}} \times \text{Cost for a matrix-vector multiply} \sim n_{\text{iter}} \times N.$$

A ***direct method*** computes a compressed representation for $(I + A)^{-1}$.

- Cost for pre-computing the inverse.
- Cost for applying the inverse to a vector.

In many environments, both of these costs can be made $O(N)$.

Direct methods are good for (1) ill-conditioned problems, (2) problems with multiple right-hand sides, (3) spectral decompositions, (4) updating, ...

Practical aspects:

Direct methods tend to be more **robust** than iterative ones.

This makes them more suitable for "black-box" implementations.

It is our empirical experience that commercial software developers seem to avoid implementing iterative solvers whenever possible (for good reasons or not).

The effort to develop direct solvers should be viewed as a step towards getting an LAPACK-type environment for solving the basic linear boundary value problems of mathematical physics.

Sampling of related work:

1996 scattering problems, *E. Michielssen, A. Boag and W.C. Chew*,

1998 factorization of non-standard forms, *G. Beylkin, J. Dunn, D. Gines*,

1998 $\mathcal{H}$-matrix methods, *W. Hackbusch, et al*,

2002 inversion of Lippmann-Schwinger equations, *Y. Chen*.

Current state of the research

The fast direct solvers we are developing exploit the fact that off-diagonal blocks of the matrix to be inverted have low rank.

This restricts the range of application to non-oscillatory, or moderately oscillatory problems. In other words, we **can** do:

- Laplace's equation, equations of elasticity, Yukawa's equation,...
- Helmholtz' and Maxwell's equations for low- and intermediate freqs.

(In special cases, high frequency problem can also be solved.)

Boundary integral equations in 2D are completely understood.

Lippmann-Schwinger eqns in 2D are well understood, implementation is under way.

In 3D, we "know" how to solve the problem, but much work remains.

Technical aspects:

Once you have a compressed inverse, applying it is very similar to applying the original operator using an FMM:

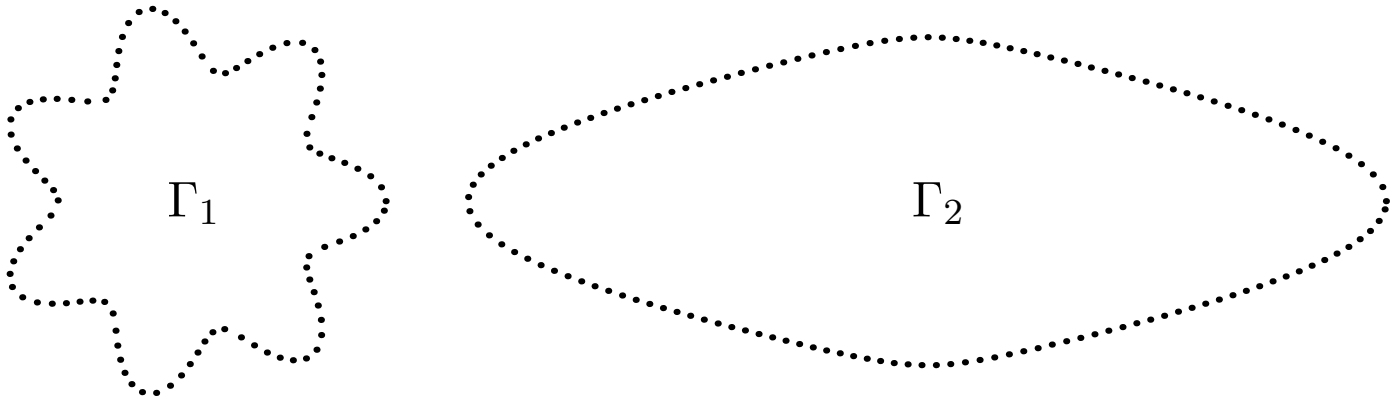
- Hierarchical (adaptive) subdivision of the computational domain.
- Outgoing fields are aggregated through an upwards pass.
- Incoming fields are aggregated through a downwards pass.

Pre-computation of the inverse operator is slightly different:

- There is only an upwards pass.
- For each subdomain, we compute operators, rather than fields. (Inverses, and "Schur complements").

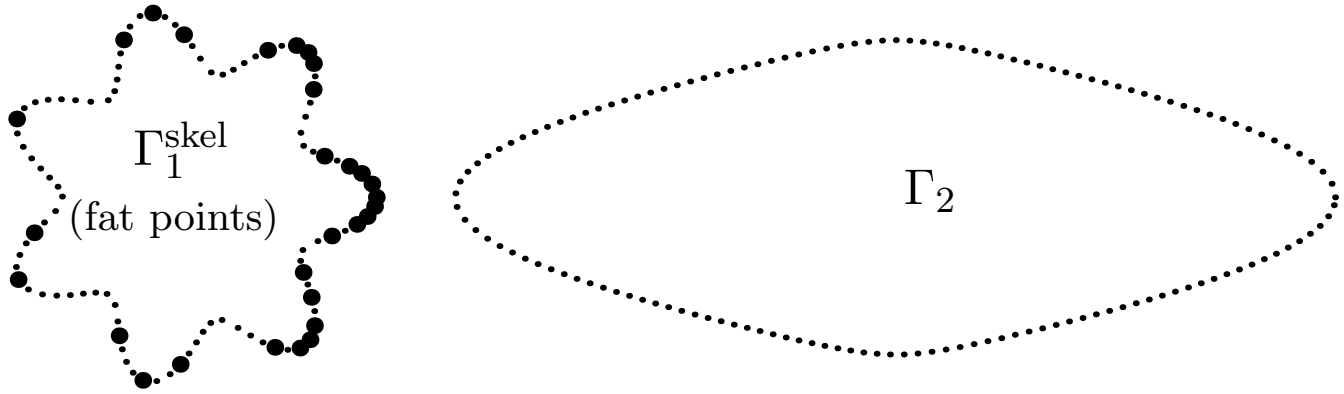
An important difference between the current methods and classical FMM:

> The outgoing and incoming fields are represented by **tabulation**, rather than via expansions in function series.



100 charges on Γ_1 induce a potential v on Γ_2.

The same potential v can be reproduced by placing only 30 charges on the bold points in the figure below. (To within precision 10^{-10}.)



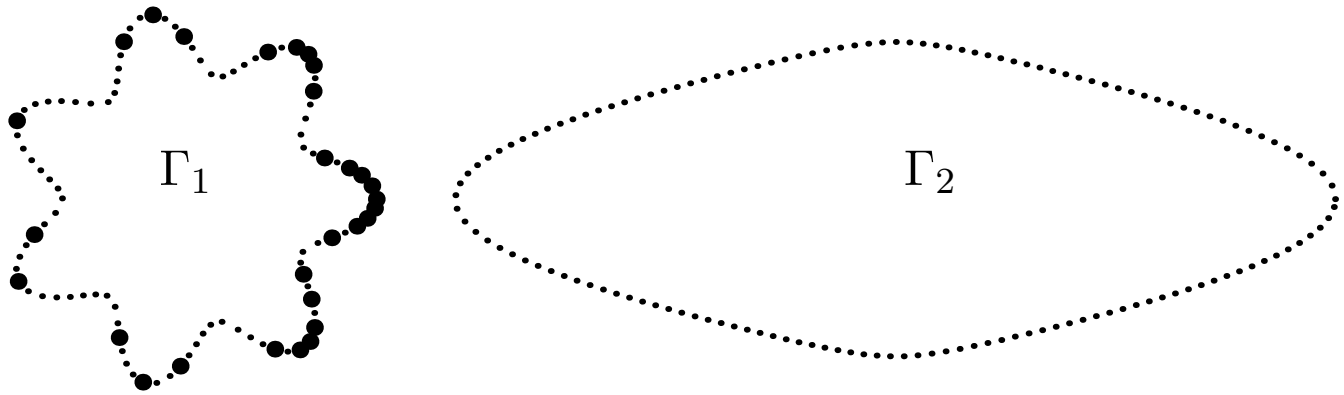
One can pick the 30 charge locations so that they work for ***any*** *distribution of charges.*

(In the classical FMM, a multipole expansion was used to reproduce v.)

An analogous representation exists for **incoming** potentials.

Let w be a potential on Γ_1 caused by a charge distribution on Γ_2.

It is possible to pick 30 points on Γ_1 in such a way that if w is known on these 30 points, then through a local interpolation, w can be determined at the remaining points.



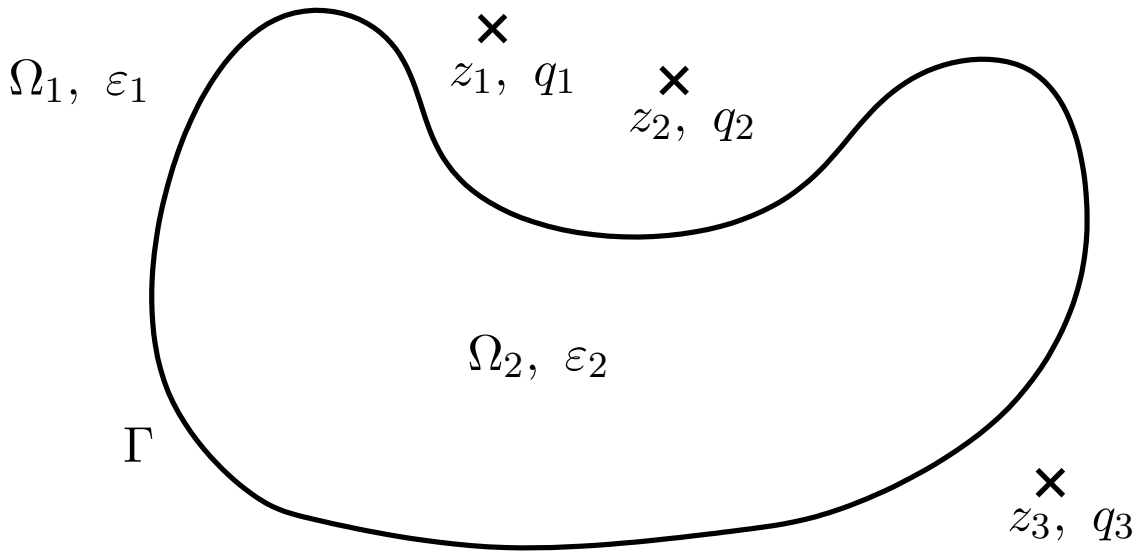
(In the classical FMM, the incoming potentials are recorded by keeping track of expansions in harmonic polynomials, or, better, exponentials.)

Numerical examples

The algorithm was implemented in Matlab and FORTRAN 77.

The experiments were run on a Pentium IV with a 2.8Ghz processor and 512 Mb of RAM.

Example 1: Electro-statics in a bi-phase dielectric medium



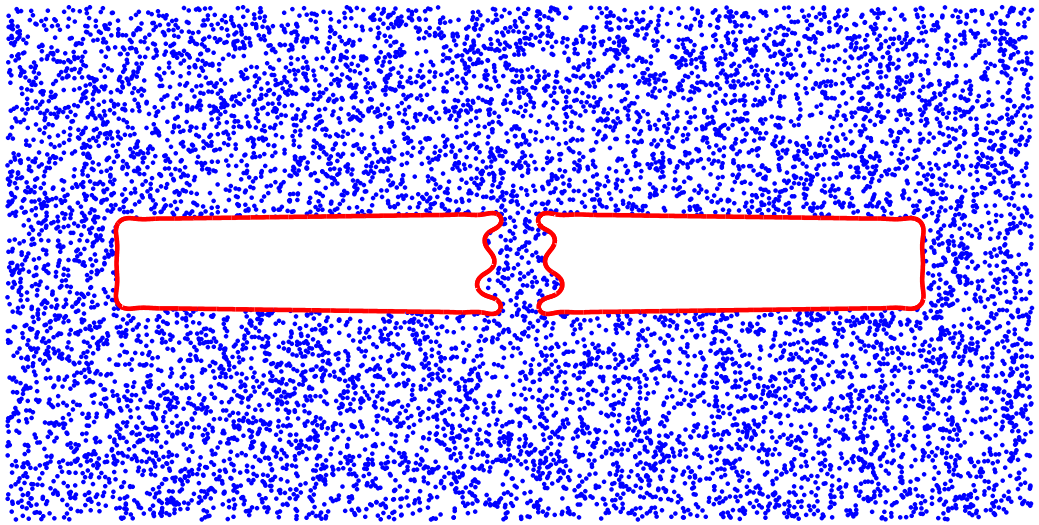
The domain Ω_2 has a different dielectric constant than the background.

There are electric charges at the points $z_i \in \Omega_1$.

Task: Evaluate all electro-static forces on the point-charges.

The "induced" charge distribution σ on Γ satisfies, for $x \in \Gamma$,

$$\frac{\varepsilon_1 + \varepsilon_2}{2(\varepsilon_1 - \varepsilon_2)}\sigma(x) - \frac{1}{2\pi}\int_\Gamma \frac{n(y)\cdot(x-y)}{|x-y|^2}\sigma(y)\,ds(y) = -\varepsilon_1 \frac{\partial v}{\partial n}(x),$$



Computational accuracy $= 10^{-10}$.

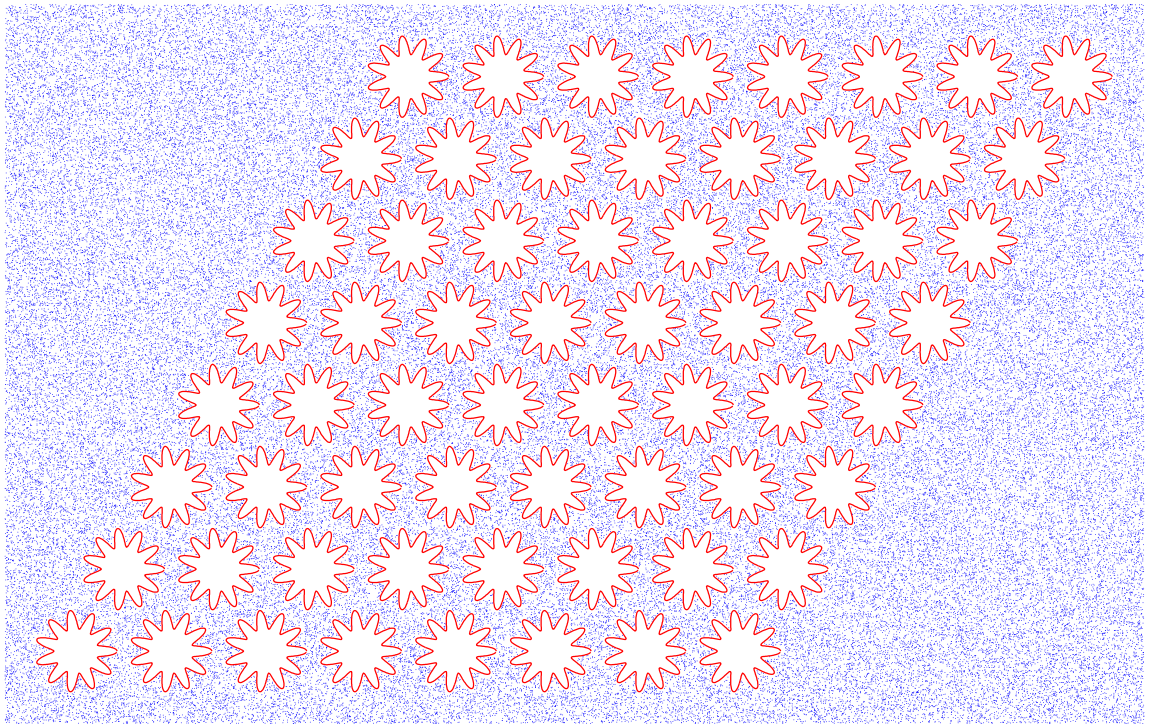
Nr. of points in the discretization of the contour $= 25\,600$.

Nr. of particles $= 10\,000$.

Time to invert the boundary integral equation $= 2.9$sec.

Time to compute the induced charges $= 0.034$sec. (2.0sec for the FMM)

Total time for the electro-statics problem $= 0.65$sec.

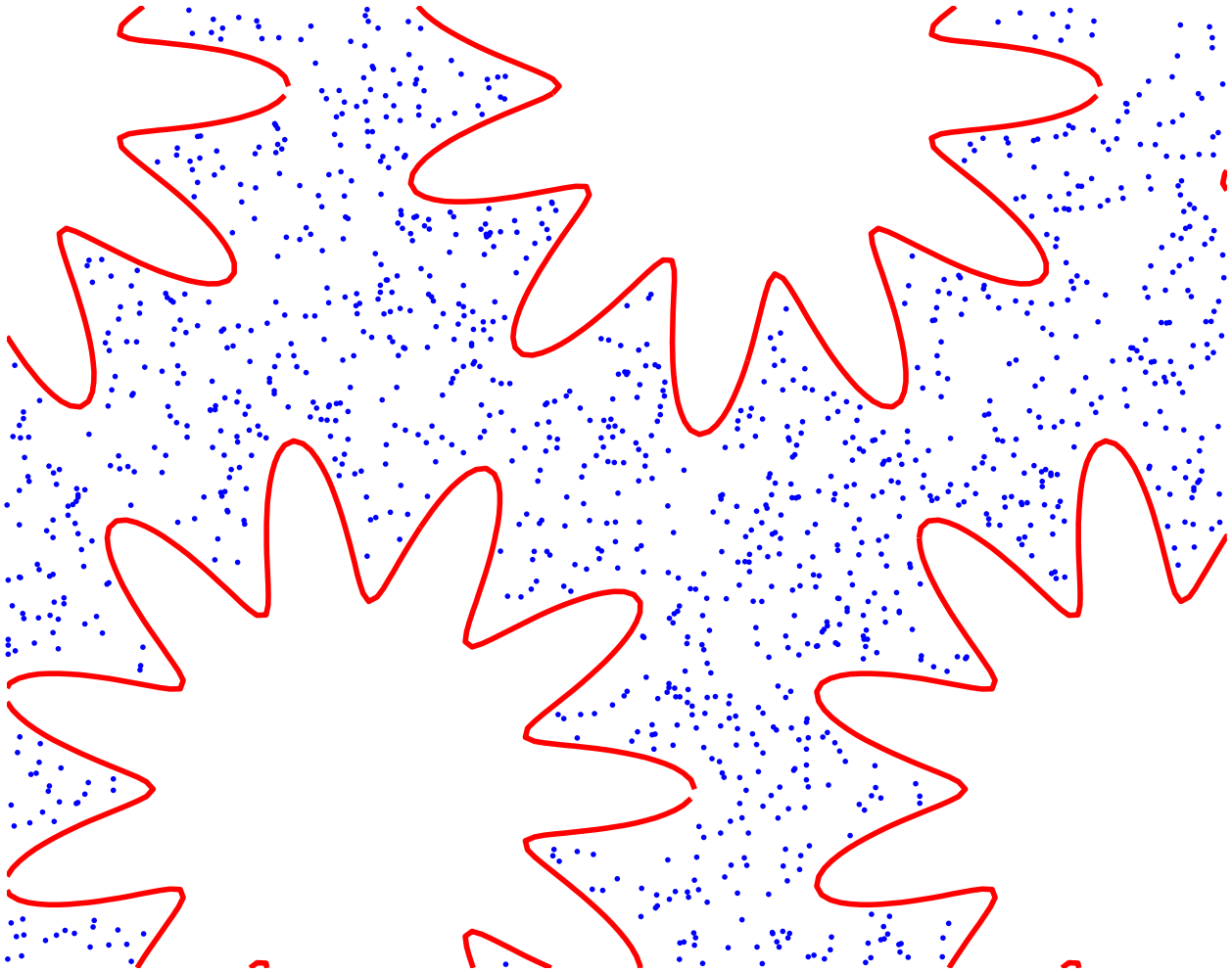


$\varepsilon = 10^{-5}$ $N_{\text{contour}} = 25\,600$ $N_{\text{particles}} = 100\,000$

Time to invert the boundary integral equation = 46sec.

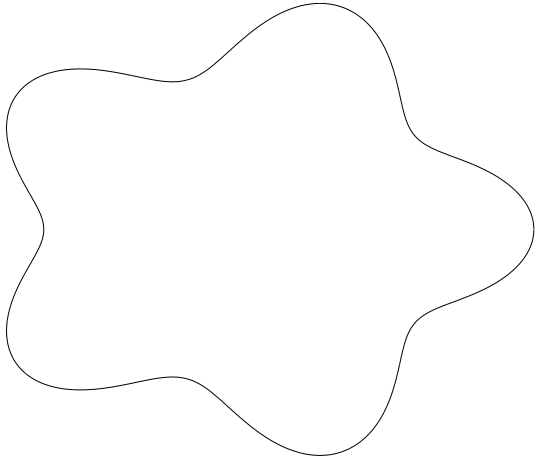
Time to compute the induced charges = 0.42sec.(2.5sec for the FMM)

Total time for the electro-statics problem = 3.8sec.



A close-up of the particle distribution.

Example 2 - An interior Helmholtz Dirichlet problem

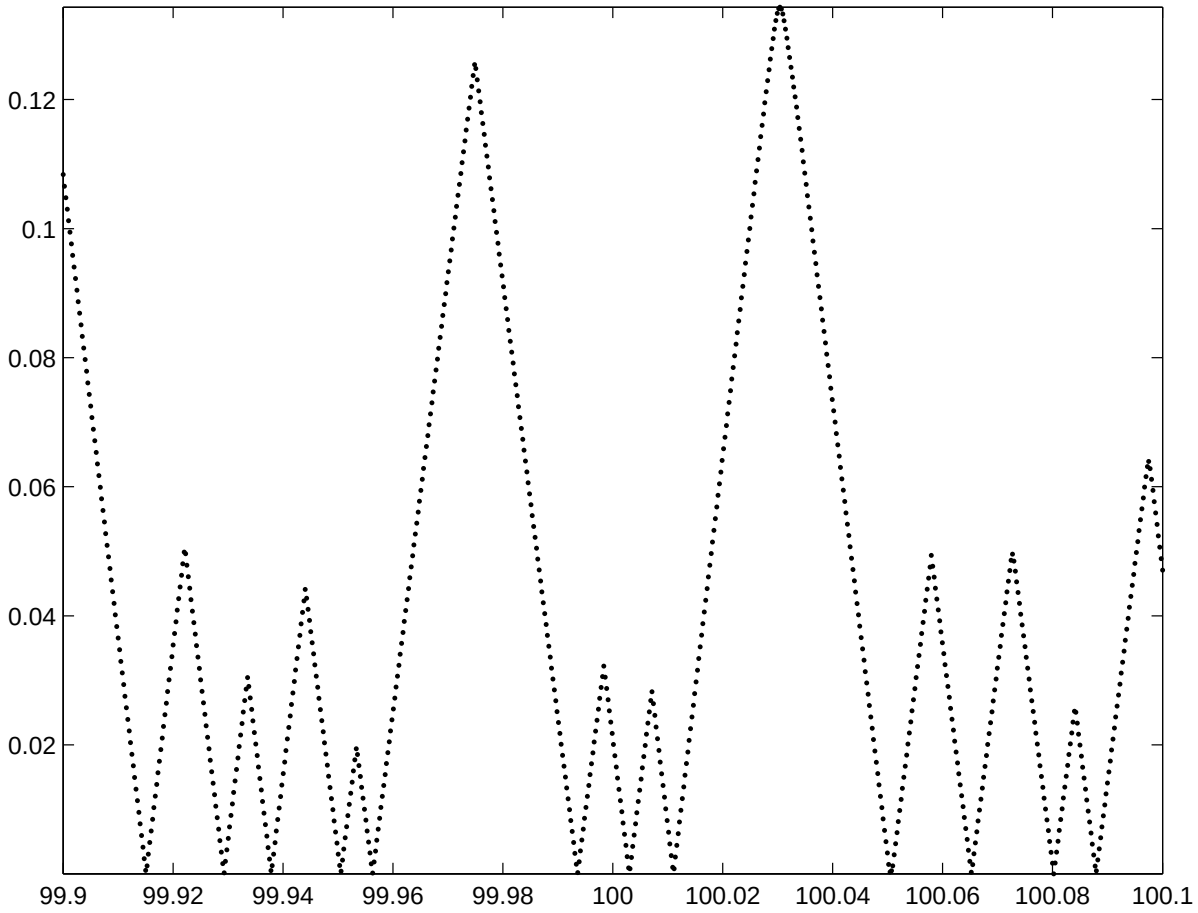


The diameter of the contour is about 2.5. An interior Helmholtz problem with Dirichlet boundary data was solved using $N = 6\,400$ discretization points, with a prescribed accuracy of 10^{-10}.

For $k = 100.011027569\cdots$, the smallest singular value of the boundary integral operator was $\sigma_{\min} = 0.00001366\cdots$.

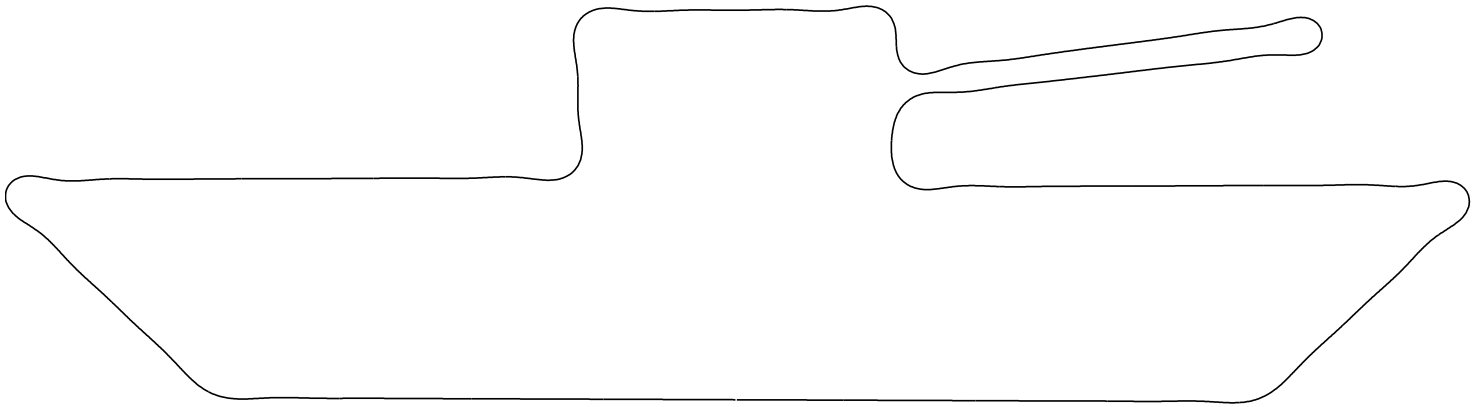
Time for constructing the inverse: 0.7 seconds.

Error in the inverse: 10^{-5}.



Plot of $\sigma_{\min}$ versus k for an interior Helmholtz problem on the smooth pentagram. The values shown were computed using a matrix of size $N = 6400$. Each point in the graph required about $60s$ of CPU time.

Example 3 - An exterior Helmholtz Dirichlet problem

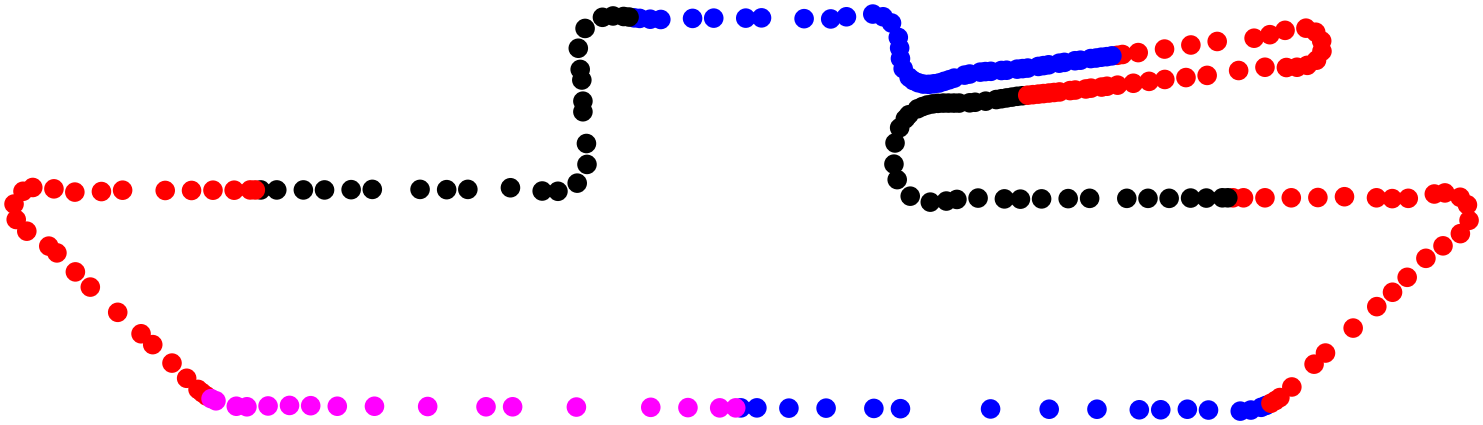


A smooth contour. Its length is roughly 15 and its horizontal width is 2.

k	N_{start}	N_{final}	t_{tot}	t_{solve}	E_{res}	E_{pot}	σ_{min}	M
21	800	435	1.5e+01	3.3e-02	9.7e-08	7.1e-07	6.5e-01	12758
40	1600	550	3.0e+01	6.7e-02	6.2e-08	4.0e-08	8.0e-01	25372
79	3200	683	5.3e+01	1.2e-01	5.3e-08	3.8e-08	3.4e-01	44993
158	6400	870	9.2e+01	2.0e-01	3.9e-08	2.9e-08	3.4e-01	81679
316	12800	1179	1.8e+02	3.9e-01	2.3e-08	2.0e-08	3.4e-01	160493
632	25600	1753	4.3e+02	7.5e+00	1.7e-08	1.4e-08	3.3e-01	350984

Computational results for an exterior Helmholtz Dirichlet problem discretized with 10^{th} order accurate quadrature. The Helmholtz parameter was chosen to keep the number of discretization points per wavelength constant at roughly 45 points per wavelength (resulting in a quadrature error about 10^{-12}).

Note: For this problem, the complexity is $O(n + (Lk)^3)$.



The nodes left at an intermediate level.

Notice the inherent adaptivity of the procedure.

(The figure actually shows the results of a Laplace problem.)

Example 4: High-frequency forwards scattering from an elongated scatterer

We solve a scalar scattering problem (Helmholtz' equation) on the wave pattern above.

The wave-length of the physical waves is roughly the same as that for the radiating waves.

The contour is discretized into 70 points per wave-length.

n_{tot}	n_{wave}	t_{comp}	t_{apply}	E_{res}	M
800	11	1.07e0	2.20e-3	9.2e-7	2.6e0
1600	23	2.32e0	4.70e-3	1.8e-7	5.5e0
3200	46	4.86e0	9.60e-3	7.7e-7	1.1e1
6400	91	1.12e1	1.94e-2	6.2e-7	2.3e1
12800	183	2.30e1	3.90e-2	8.8e-7	4.7e1
25600	366	4.58e1	7.92e-2	4.7e-6	9.4e1
51200	731	1.01e2	1.59e-1	4.9e-6	1.9e2
102400	1463	2.02e2	3.19e-1	6.3e-6	3.8e2
204800	2926	4.06e2	6.36e-1	1.7e-5	7.6e2

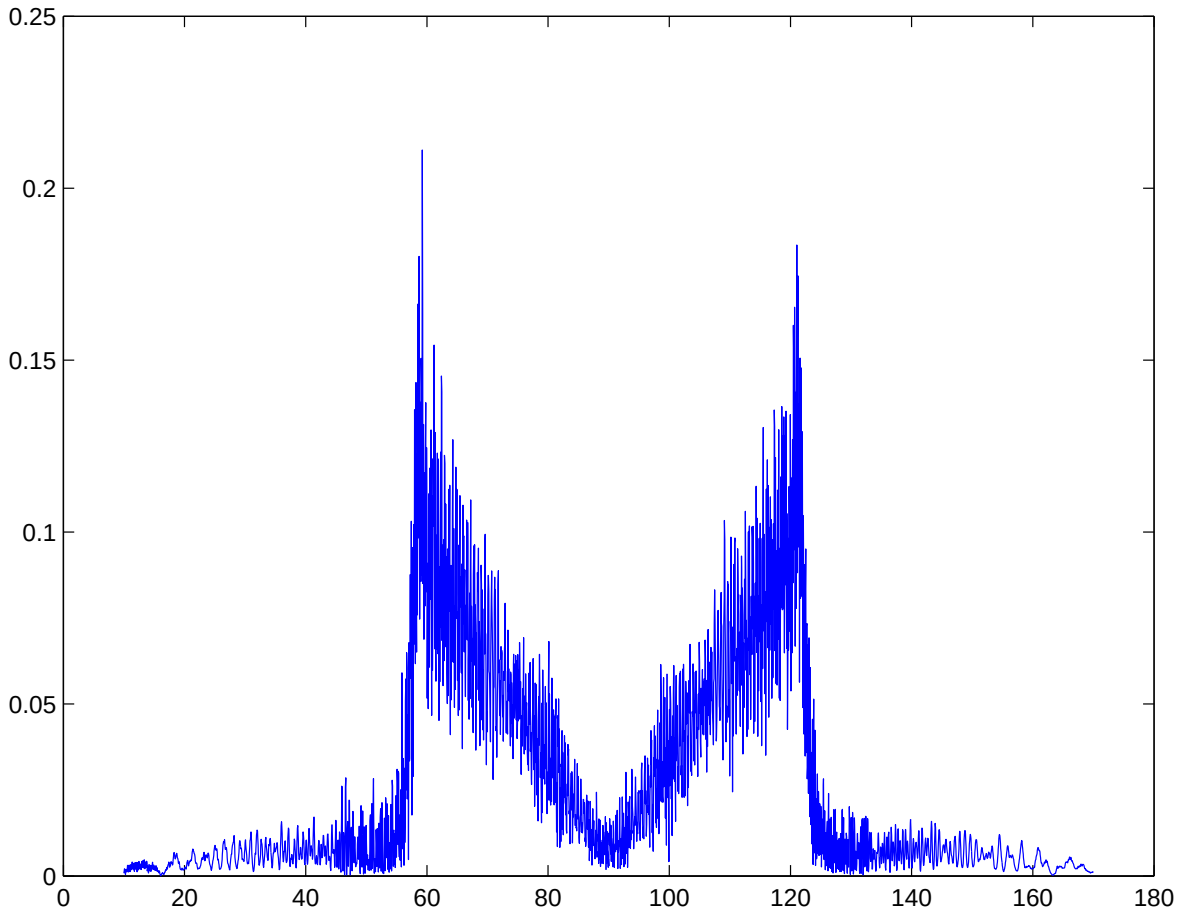
n_{wave} is the size of the scatterer in wavelengths.

t_{comp} is the time (in seconds) required to invert the matrix.

t_{apply} is the time (in seconds) required to apply the inverse.

E_{res} is the (relative) error in the residual.

M is the amount of memory required (in Mb).



The reflection profile from a surface with 600 waves.

But what about iterative solvers?

Well, there is good news; the tabulation technique developed for the direct solvers can also be used to accelerate matrix-vector multipliers.

For **1D** problems (including contour integral equations), we have achieved major speed-ups over the fastest existing FMMs.
Break-even point with direct computation at double precision accuracy is less than 100.
This has implications for numerical linear algebra.

For **2D** problems (including surface integral equations in space), there is some speed-up, and significant simplifications in the implementation. As an example, anisotropic elasticity will become manageable.

(For volume integral equation in **3D**, FMM based on exponential expansions will still be faster.)

Summary

Direct solvers are currently being developed for integral equations involving non-oscillatory, or moderately oscillatory, kernels.

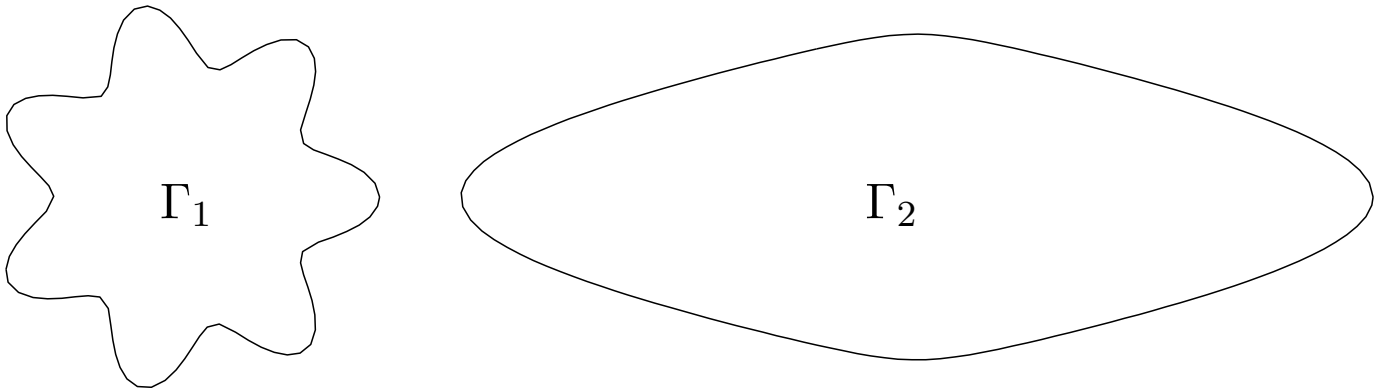
Advantages of direct solvers over iterative solvers:

- Less sensitive to the conditioning of a problem.
- Robustness.
- Computing spectral decompositions.
- Advantage in speed for problems with multiple right hand sides.

2D:	• Well-understood. • Fairly efficient implementations exist. • Direct solvers as fast as or faster than iterative solvers.
3D:	• Under development. • Quite complex machinery. • Will probably be expensive.

Other applications – direct solvers:

- Inversion of general Toeplitz matrices in $O(N)$ time.
- Compute conformal mappings in $O(N)$ time
  (by solving the Kerzman-Stein integral equations).
- Multi-scale model reduction – numerical homogenization.
- Wave-propagation in solids – nearly resonant problems.



Consider the integral equation

$$\frac{1}{2}u(x) + \int_\Gamma K(x,y)u(y)\,ds(y) = f(x), \qquad x \in \Gamma, \tag{3}$$

where $\Gamma = \Gamma_1 + \Gamma_2$ and K is the Laplace double layer kernel. Discretizing (3) using a Nyström method we obtain the system of linear equations

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix},$$

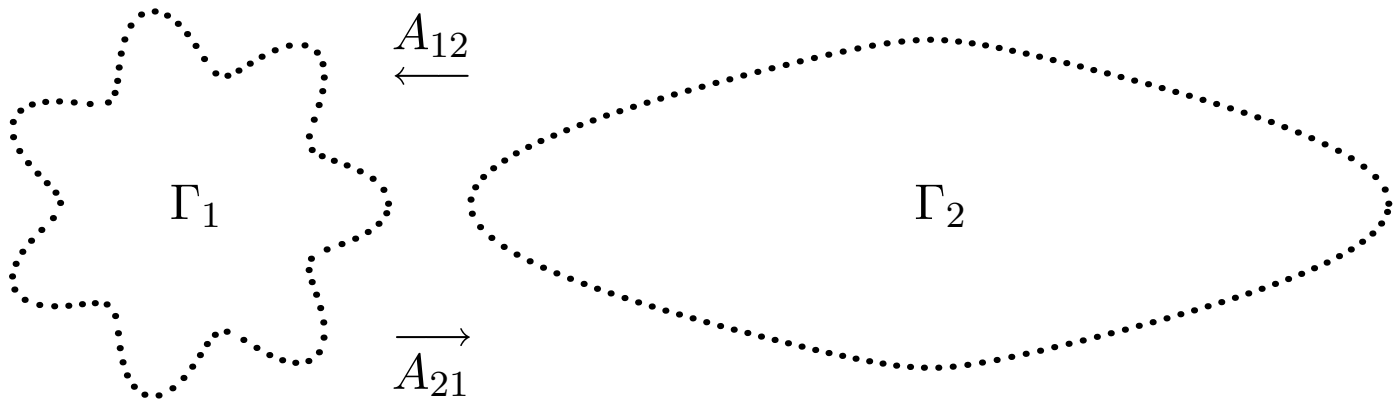
where, for $i, j = 1, 2$, the matrix A_{ij} discretizes the operator

$$\delta_{ij}\frac{1}{2}u(x) + \int_{\Gamma_j} K(x,y)\,u(y)\,ds(y), \quad x \in \Gamma_i.$$

In other words, A_{12} maps a charge distribution on Γ_2 to a potential on Γ_1.

We recall a formula for the inverse of a 2×2 block-matrix:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{-1} = \begin{bmatrix} (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} & \bullet \\ \bullet & (A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1} \end{bmatrix}.$$

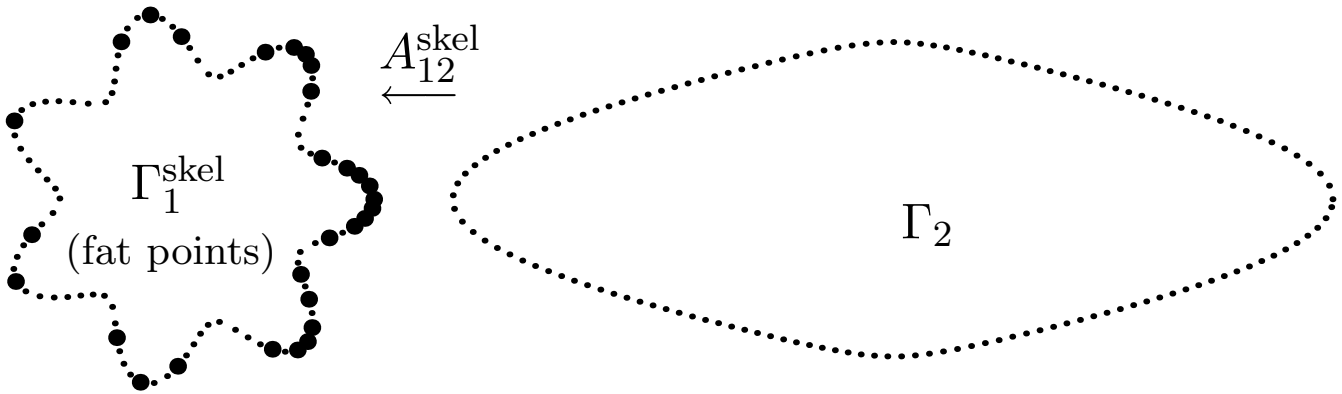


The physical meaning of the term $A_{21}A_{11}^{-1}A_{12}$ in the formula for A^{-1}:

$$\text{Charges on } \Gamma_2 \xrightarrow{A_{12}} \text{Pot. on } \Gamma_1 \xrightarrow{A_{11}^{-1}} \text{Charges on } \Gamma_1 \xrightarrow{A_{21}} \text{Pot. on } \Gamma_2$$

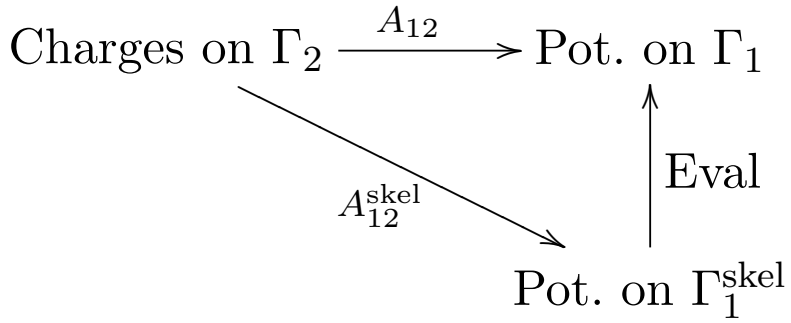
The maps A_{12} and A_{21} are typically rank-deficient (to finite precision).

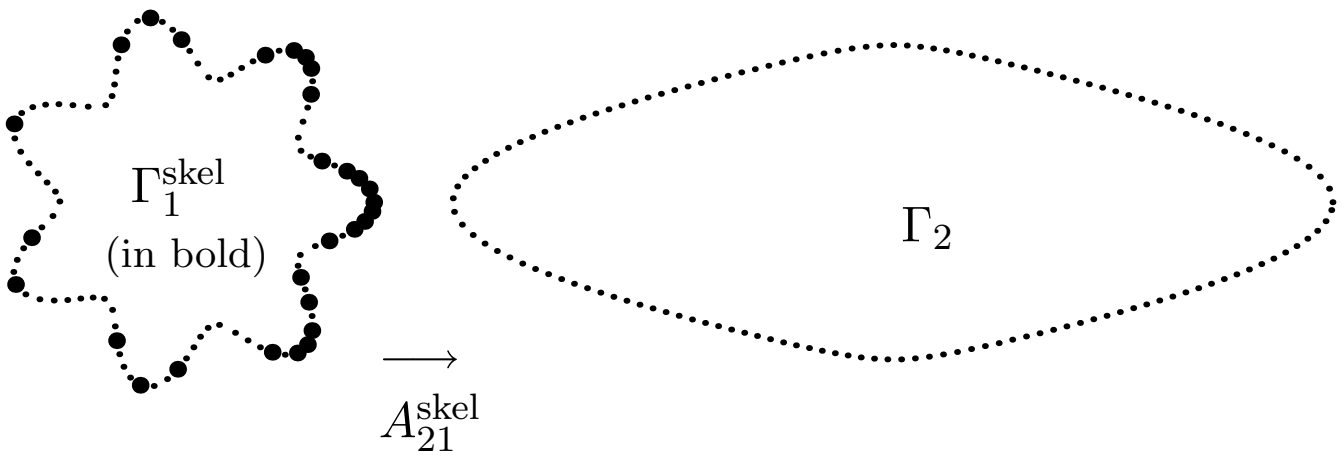
Example: Laplace double layer kernel: to accuracy 10^{-10}, the rank is 30.



Let k denote the rank of A_{12} (to precision ε).

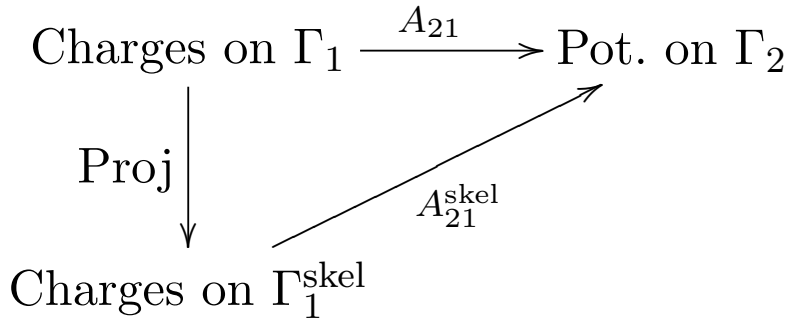
There exists a set $\Gamma_1^{\text{skel}} \subset \Gamma_1$ with k points and a map Eval such that the following diagram commutes (to precision ε).

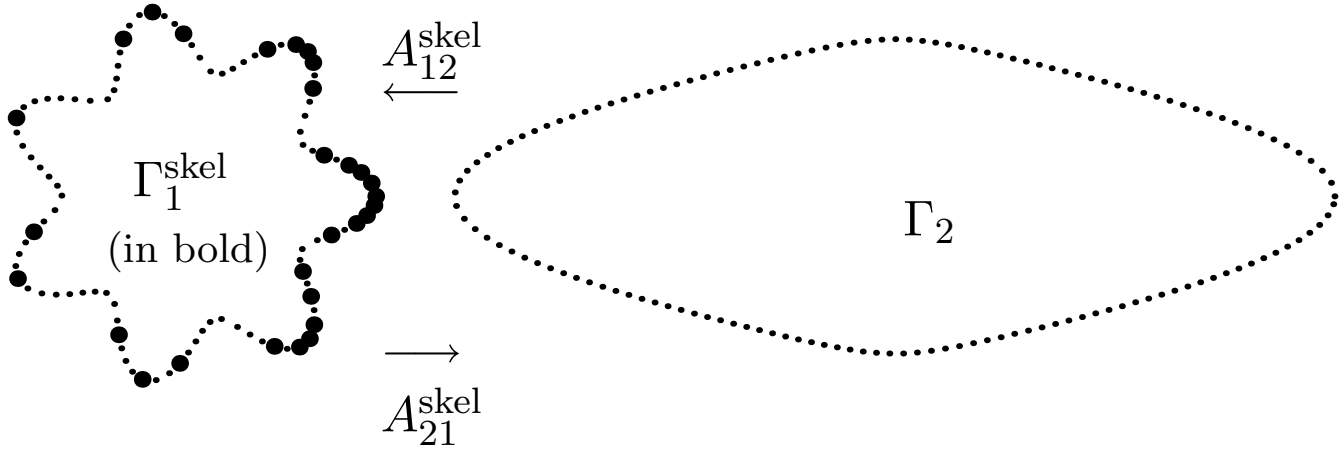




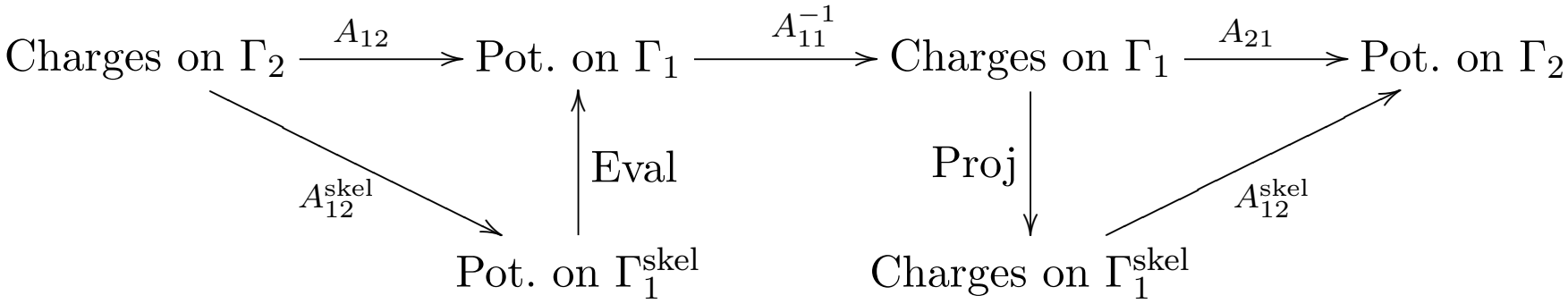
Analogously, we can compress A_{21}:

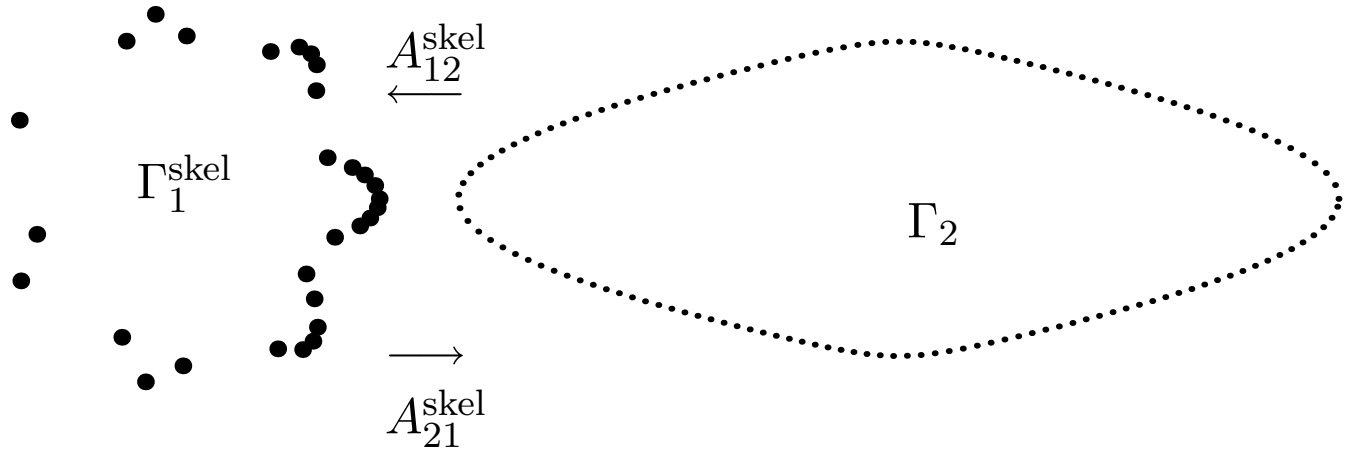
There exists a set $\Gamma_1^{\text{skel}} \subset \Gamma_1$ with k points and a map Proj such that the following diagram commutes (to precision ε).



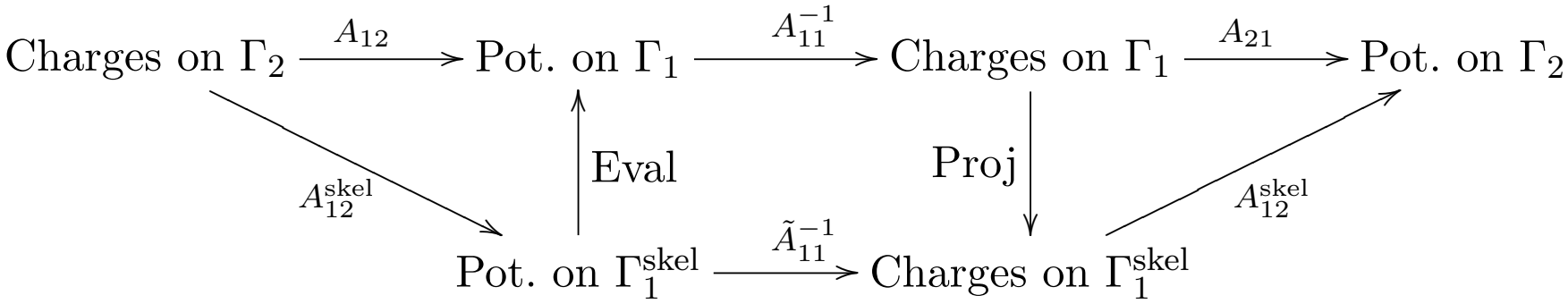


Now we can compress the entire interaction...





. . . and completely forget about the original points!



Notes:

- A_{12}^{skel} consists of k of the rows of A_{12}.
- A_{21}^{skel} consists of k of the columns of A_{21}.
- The process consists of **pure linear algebra**.
- Proven to be accurate and well-conditioned.
  - Gu and Eisenstat (SIAM J. Sci. Comp. 1996)
  - Cheng, Gimbutas, Martinsson, Rokhlin (SIAM J. Sci. Comp. 2005)
  - Martinsson and Rokhlin (J. Comp. Phys. 2005)