Introduction to Integral Equation Methods, Fast Algorithms, and the Hadamard Conjecture

Shidong Jiang

New Jersey Institute of Technology

January 3, 2017

HKUST-ICERM VI-MSS Workshop

Part I. Integral Equation Methods

Basic observation

Many problems originally posed in the PDE form can be solved as follows.

- Convert it to an integral equation formulation that is as well-conditioned as the underlying physical problem.
- Discretize the involved integrals accurately via some high-order quadrature.
- Solve the resulting linear system and/or evaluate the resulting discrete summation efficiently via fast algorithms.

Integral equation formulation and potential theory

- For many problems, the starting point is the so-called fundamental solution or Green's function G for a constant coefficient differential operator L. The original PDE does not have to be constant coefficient.
- To deal with the inhomogeneous term, one may need the so-called volume potential, which is the convolution integral of G with a function f on the entire domain.
- To deal with the boundary conditions, one needs various layer potentials that are the convolution of G or its partial derivatives with a function σ on the boundary.
- For time-dependent problems, one also needs the so-called initial potential that is the convolution of G with the initial data u_0 on the spatial domain.

Green's function for the Laplace/Poisson equation

In the following, both x and y are points in $\mathbb{R}^n$; all differential operators are acting on x, y is regarded as a fixed source point. δ refers to the Dirac delta function whose rigorous definition requires the theory of distributions (or generalized functions) developed by Laurent Schwartz in 1940s with proper test functions (for example, $C_0^\infty(\mathbb{R}^n)$, i.e., smooth functions with compact support).

- The Laplace equation

$$\Delta G(x,y) = -\delta(x-y)$$

$$G(x,y) = \begin{cases} -\dfrac{1}{2\pi}\log r, & n = 2 \\ \dfrac{1}{(n-2)\omega_n}\dfrac{1}{r^{n-2}}, & n > 2. \end{cases} \tag{1.1}$$

Here $r = |x-y|$, $\omega_n = 2\sqrt{\pi}^n/\Gamma(\frac{1}{2}n)$ is the surface area of the unit sphere in $\mathbb{R}^n$. In particular, $\omega_2 = 2\pi$, $\omega_3 = 4\pi$.

Green's function for the Helmholtz equation

- The Helmholtz equation

$$(\Delta + k^2)G(x,y) = -\delta(x-y)$$

$$G(x,y) = \begin{cases} \frac{i}{4} H_0^{(1)}(k|x-y|), & n=2 \\ \frac{1}{4\pi}\frac{e^{ik|x-y|}}{|x-y|}, & n=3. \end{cases} \tag{1.2}$$

Here $H_0^{(1)}$ is the zeroth order Hankel function of the first kind (see, for example, NIST Digital Library of Mathematical Functions at `http://dlmf.nist.gov/`). The Green's function satisfies the Sommerfeld radiation condition:

$$\lim_{r\to\infty} r^{\frac{n-1}{2}} \left(\frac{\partial}{\partial r} - ik \right) G(r) = 0.$$

The Poisson equation in free space

The Poisson equation

$$-\Delta u(x) = f(x), \qquad x \in \mathbb{R}^3, \qquad \lim_{|x|\to\infty} u(x) = 0. \tag{1.3}$$

And the solution to (1.3) is simply the volume potential.

$$u(x) = V[f](x) := \int_{\mathbb{R}^3} \frac{1}{4\pi|x-y|} f(y)\, dy, \qquad x \in \mathbb{R}^3. \tag{1.4}$$

The Fourier transform of the G is $\widehat{G}(k) = \int_{\mathbb{R}^3} G(x)\, e^{-i\,k\cdot x}\, dx = \frac{1}{|k|^2}$. Thus when f is smooth and has compact support, $u(x)$ can be computed via nonuniform fast Fourier transform efficiently and accurately (Jiang, Greengard, and Bao 2012). In fact, one could replace G by $\tilde{G}(x) = G(x)\mathcal{I}_{B_R(0)}$, where $\mathcal{I}_{B_R(0)}$ is the indicator function of $B_R(0)$. It is straghtforward to calculate the Fourier transform of $\tilde{G}(x)$ and show that it is actually smooth. One may then simply use the FFT to compute $u(x)$ to high accuracy.

When f is highly nonuniform, one may switch to the fast multipole method to compute $u(x)$ directly.

Other PDEs in free space

- The same approach can be applied to solve many other PDEs in free space including the Helmholtz equation $\Delta u + k^2 u = f$, the Yukawa equation or the Poisson-Boltzmann equation $\Delta u - k^2 u = f$, the biharmonic equation $\Delta^2 u = f$, etc.
- It can also be used to solve the so-called fractional PDEs. For example, The fractional Poisson equation

$$(-\Delta)^{1/2} u(x) = f(x), \qquad x \in \mathbb{R}^2, \qquad \lim_{|x|\to\infty} u(x) = 0. \tag{1.5}$$

  The Green's function is $G(x, y) = \frac{1}{2\pi|x-y|}$ and the solution to (1.5) is again given by the volume potential

$$u(x) = \int_{\mathbb{R}^2} \frac{1}{2\pi|x-y|} f(y)\, dy, \qquad x \in \mathbb{R}^2. \tag{1.6}$$

- Once again, one may use the FMM to evaluate $u(x)$ when f is highly nonuniform and simply the FFT when a uniform grid is sufficient to resolve f. Another approach is to approximate $1/r$ by a sum of Gaussians.

Performance comparison

- The multigrid method is the most efficient way to solve the elliptic PDEs directly.
- For a detailed comparison of the performance of various schemes, see the paper "FFT, FMM, or multigrid? A comparative study of state-of-the-art Poisson solvers for uniform and nonuniform grids in the unit cube", by A. Gholami, D. Malhotra, H. Sundar, and G. Biros in SIAM J. Sci. Comput. **38**, no. 3, C280–C306, 2016.
- One important remark is that here the FMM refers to the *volume* FMM instead of the classical FMM for computing the discrete summations. That is, the FMM here precomputes and stores tables for all the integrals involved the near interactions and the tree has to be a level restricted $2:1$ tree.

Boundary value problems

- Suppose we are trying to solve the Dirichlet problem of the Poisson equation:

$$\begin{aligned} -\Delta u(x) = f(x), &\qquad x \in D \\ u = g(x). &\quad x \in \partial D. \end{aligned} \tag{1.7}$$

- The inhomogeneous term is easily dealt with the volume potential

$$V[f](x) = \int_D G(x, y) f(y) dy. \tag{1.8}$$

  Here the density is simply f and one only needs to evaluate the volume potential. There is no need to solve!!

- The boundary condition will be dealt with layer potentials with some unknown density.
- By the superposition principle for linear problems, the whole representation could look like this:

$$u(x) = V[f](x) + D[\sigma](x). \tag{1.9}$$

Layer potentials

- Let $D \subset \mathbb{R}^n$ be a bounded domain. Its boundary is denoted by ∂D with ν the unit outward normal vector.
- The single layer potential is defined by the formula:

$$S[\sigma](x) = \int_{\partial D} G(x, y)\sigma(y) ds_y. \tag{1.10}$$

- The double layer potential is defined by the formula:

$$D[\sigma](x) = \int_{\partial D} \frac{\partial G(x, y)}{\partial \nu(y)}\sigma(y) ds_y. \tag{1.11}$$

- Higher-order layer potentials such as the quadruple layer potential, the octuple layer potential, etc. can be defined similarly by taking higher-order derivatives of the Green's function as the kernels.

Single and Double Layer Potentials

- Very often we may also need to consider the normal derivatives of single and double layer potentials.

$$S'[\sigma](x) = \frac{\partial}{\partial \nu(x)} \int_{\partial D} G(x,y)\sigma(y) ds_y. \tag{1.12}$$

$$D'[\sigma](x) = \frac{\partial}{\partial \nu(x)} \int_{\partial D} \frac{\partial G(x,y)}{\partial \nu(y)}\sigma(y) ds_y. \tag{1.13}$$

Jump Relations

- Assume for now that the boundary ∂D is of class C^2.
- The single layer potential is continuous across the boundary, while its normal derivative satisfies the following jump relation:

$$\lim_{\varepsilon \to 0+} S'[\sigma]\,(x \pm \varepsilon \nu_x) = S'[\sigma]\,(x) \mp \frac{1}{2}\sigma(x), \quad x \in \partial D. \quad (1.14)$$

- The double layer potential satisfies the following jump relation:

$$\lim_{\varepsilon \to 0+} D[\sigma]\,(x \pm \varepsilon \nu_x) = D[\sigma]\,(x) \pm \frac{1}{2}\sigma(x), \quad x \in \partial D, \quad (1.15)$$

while its normal derivative D' is actually continuous across the boundary.

What about the Helmholtz equation?

- The jump relations of the layer potentials for the Helmholtz equation are exactly the same as those for the Laplace equation.
- This is because the leading singular term of the Green's function of the Helmholtz equation is identical to the Green's function of the Laplace equation.

Another useful fact

- A well-known fact in potential theory states that

$$(x-y)\cdot\nu_y = O(|x-y|^2), \text{ for } y \to x \text{ when } x, y \in \partial D. \quad (1.16)$$

  Hence,

$$\frac{\partial G(x,y)}{\partial \nu_y} = \frac{1}{4\pi}\frac{(x-y)\cdot\nu_y}{|x-y|^3} = O\left(\frac{1}{|x-y|}\right). \qquad (1.17)$$

- That is, the kernel of the double layer potential is only weakly singular and thus integrable.

Boundary integral equations

- For the Dirichlet problem, we represent the solution via the double layer potential and need to solve the following boundary integral equation (BIE):

$$\pm\frac{1}{2}\sigma(x) + D[\sigma](x) = b(x). \tag{1.18}$$

- For the Neumann problem, we represent the solution via the single layer potential and need to solve the following boundary integral equation:

$$\mp\frac{1}{2}\sigma(x) + S'[\sigma](x) = b(x). \tag{1.19}$$

- Similarly, the Robin problem can be solved via a single layer potential representation.

Advantages of the BIE approach

- The unknown density σ is only on the boundary. This reduces the dimension of the problem by one in the solve phase and thus the total number of unknowns by a large extent.
- It is easier to design high-order discretization scheme for the boundary and the unknowns on the boundary rather than the whole volume and the unknowns in the whole volume, especially in the case of complex geometries. In two dimensions, one only needs to discretize the boundary curves instead of the 2D domain; while in three dimensions, one only needs to discretize the boundary surfaces instead of the 3D domain.
- The integral formulation leads to a well-conditioned linear system which requires a constant number of iterations to solve. This is only true for the so-called second kind integral equations.
- For exterior problems, there is no need to design artificial boundary conditions to truncate the computational domain when the integral formulation is used.

Types of integral equations

- Volterra - the integration domain depends on the target point. Fredholm - the integration domain is the whole boundary.
- First kind - the integral operator is compact.
- Second kind - the integral operator is Identity operator + Compact operator.
- Singular - the integral operator is singular, i.e., the integrals are defined in the sense of principal value. Examples include the Hilbert transform in 1D, the Riesz transform in higher dimensions, and layer potentials with the tangential derivative of the Green's function as the kernel.
- Hypersingular - the integral operator is hypersingular, i.e., the integrals are defined in the sense of finite part. Examples include the layer potentials with the second order derivatives of the Green's function as the kernel for second order PDEs.

Integral operators

- Integral operators - Let $G \in \mathbb{R}^m$ be a nonempty compact and Jordan measurable set (i.e., its characteristic function ξ_G is Riemann integrable). Then the linear operator $A : X(G) \to X(G)$ defined by
$$A[\phi](x) := \int_G K(x,y)\phi(y)dy, \quad x \in G,$$
is called an integral operator. Here $K : G \times G \to \mathbb{C}$ is the kernel of the operator.
- X could be C (space of continuous functions), $C^{0,\alpha}$ (Holder continuous functions with exponent α), L^2 (also a Hilbert space when combined with the natural inner product), L^p, H^P (Sobolev space, roughly speaking, it contains functions whose p-th order weak derivative also belongs to L^2).
- K could be continuous, weakly singular, singular, or hypersingular.
- The associated integral operator A might be compact, compact, bounded, and unbounded.

Compact operators

- A linear operator $A : X \to Y$ is called compact if it maps each bounded set in X into a relatively compact set in Y.
- A is compact if and only if for each bounded sequence (ϕ_n) in X the sequence $(A\phi_n)$ contains a convergent subsequence in Y.
- Compact linear operators are bounded.
- If A and B are compact, then their linear combination $\alpha A + \beta B$ is also compact.
- If A and B are bounded, then AB is compact if one of them is compact.
- If A is the limit of a sequence compact operators (A_n), then A is compact.
- A bounded operator with finite dimensional range is compact.
- An integral operator with continuous or weakly singular kernel is compact.

Riesz Theory

- Let I be the identity operator and $A : X \to X$ be a compact operator. Define the second kind operator $L$by

$$L := I - A.$$

- First Riesz Theorem. The nullspace of the operator L, i.e.,

$$N(L) := \{\phi \in X : L\phi = 0\},$$

  is a finite dimensional subspace.
- Second Riesz Theorem. The range of L $L(X) := \{L\phi : \phi \in X\}$, is a *closed* linear subspace.
- Theorem. $I - A$ is injective (one-to-one) if and only if it is surjective (onto).

Second kind integral operators

- If the homogeneous equation
$$\phi - A\phi = 0$$
only has the trivial solution $\phi = 0$, then for each $f \in X$ the inhomogeneous equation
$$\phi - A\phi = f$$
has a unique solution $\phi \in X$ and this solution depends continuously on f.
- If the homogeneous equation has m linearly independent solutions $\phi_1, \cdots, \phi_m$, then the inhomogeneous is either unsolvable or its general solution is of the form
$$\phi = \tilde{\phi} + \sum_{k=1}^{m} \alpha_k \phi_k,$$
where $\tilde{\phi}$ is a particular solution.
- That is, a second kind linear operator equation behaves exactly like a finite dimensional linear system.

Spectral theory for compact operators

- λ is an eigenvalue of A if $A\phi = \lambda\phi$ for some $\phi \neq 0$.
- λ is a regular value if $(\lambda I - A)^{-1}$ exists and is bounded.
- The set of all regular values of A is called the resolvent set $\rho(A)$.
- The complement of $\rho(A)$ in $\mathbb{C}$ is called the spectrum $\sigma(A)$ and

  $$r(A) := \sup_{\lambda \in \sigma(A)} |\lambda|$$

  is called the spectral radius of A.

Spectral theory for compact operators

- Let $A : X \to X$ be a compact linear operator on an infinite dimensional normed space X. Then $\lambda = 0$ belongs to the spectrum $\sigma(A)$ and $\sigma(A) \setminus \{0\}$ consists of at most a countable set of eigenvalues with no point of accumulation except, possibly, $\lambda = 0$.
- That is, zero is the only possible limit point of the eigenvalues of A if A is compact.

Dual system, bilinear or sesquilinear forms

- bilinear or sesquilinear forms - A mapping $(\cdot, \cdot) : X \times Y \to \mathbb{C}$ is called a sesquilinear form if

  $$(\alpha_1\phi_1 + \alpha_2\phi_2, \psi) = \alpha_1(\phi_1, \psi) + \alpha_2(\phi_2, \psi),$$

  $$(\phi, \beta_1\psi_1 + \beta_2\psi_2) = \beta_1^*(\phi, \psi_1) + \beta_2^*(\phi, \psi_2).$$

  It is bilinear if β_1^*, β_2^* are replaced by β_1, β_2.

- Two normed spaces X, Y equipped with a bilinear or sesquilinear form is called a dual system and denoted by $\langle X, Y \rangle$.

- Adjoint operator. Let $\langle X_1, Y_1 \rangle$ and $\langle X_2, Y_2 \rangle$ be two dual systems. Then two operators $A : X_1 \to X_2$, $B : Y_1 \to Y_2$ are call adjoint if

  $$\langle A\phi, \psi \rangle = \langle \phi, B\psi \rangle$$

  for all $\phi \in X_1$, $\psi \in Y_2$.

The Fredholm Alternative

- Let $A : X \to X$, $B : Y \to Y$ be compact adjoint operators in a dual system $\langle X, Y \rangle$. Then either $I - A$ and $I - B$ are bijective (i.e., one-to-one and onto and thus invertible) or $I - A$ and $I - B$ have nontrivial nullspace with finite dimension

$$\dim N(I - A) = \dim N(I - B) \in \mathbb{N}$$

and the ranges are given by

$$(I - A)(X) = \{f \in X : \langle f, \psi \rangle = 0, \psi \in N(I - B)\}$$

and

$$(I - B)(Y) = \{g \in Y : \langle \phi, g \rangle = 0, \phi \in N(I - A)\}.$$

Integral equation version

- Let K be a continuous or weakly singular kernel. Then either the homogeneous integral equations

$$\phi(x) - \int_G K(x,y)\phi(y)dy = 0, \quad x \in G,$$

$$\psi(x) - \int_G K(y,x)\psi(y)dy = 0, \quad x \in G,$$

only have the trivial solutions $\phi = 0$ and $\psi = 0$ and the inhomogeneous integral equations

$$\phi(x) - \int_G K(x,y)\phi(y)dy = f(x), \quad x \in G,$$

$$\psi(x) - \int_G K(y,x)\psi(y)dy = g(x), \quad x \in G,$$

have a unique solution for every right hand side.

Integral equation version

- or the homogeneous integral equations have the same finite number $m \in \mathbb{N}$ of linearly independent solutions and the inhomogeneous integral equations are solvable if and only if the right-hand sides satisfy

  $$\int_G f(x)\psi(x)dx = 0$$

  for all solutions ψ of the homogeneous adjoint equation and

  $$\int_G \phi(x)g(x)dx = 0$$

  for all solutions ϕ of the homogeneous equation, respectively.

Key issue of the integral equation formulation

- For SKIEs, the most important theorectical issue is probably the analysis of their nullspace. If there is a nontrivial nullspace, what is the dimension of the nullspace and how do we modify the formulation to eliminate the nullspace?
- Sometimes this can be done by adding lower order terms to the integral representation.
- One could also apply a neat trick in Sifuentes, Gimbutas, and Greengard (2015) to solve a rank-deficient linear system when the dimension of the nullspace is known.

Time-dependent PDEs

- We will use the heat equation as the example.
- The fundamental solution (or Green's function) of the heat equation satisfies the equation:

$$\frac{\partial G}{\partial t} = \Delta G + \delta(x-y)\delta(t-\tau)$$

and is given by the formula

$$G(x,t;y,\tau) = \frac{1}{[4\pi(t-\tau)]^{n/2}} e^{-\frac{|x-y|^2}{4(t-\tau)}}. \qquad (1.20)$$

Various potentials associated with the heat equation

- Volume potential:

$$V[f](x,t) = \int_0^t \int_D G(x,t;y,\tau) f(y,\tau) dy d\tau.$$

- Initial potential:

$$I[g](x,t) = \int_D G(x,t;y,0) g(y) dy.$$

- Single layer potential:

$$S[\sigma](x,t) = \int_0^t \int_{\partial D} G(x,t;y,\tau) \sigma(y,\tau) ds_y d\tau.$$

- Double layer potential:

$$D[\sigma](x,t) = \int_0^t \int_{\partial D} \frac{\partial G(x,t;y,\tau)}{\partial \nu_y} \sigma(y,\tau) ds_y d\tau.$$

Jump relations of the heat layer potentials

- The heat layer potentials satify exactly the same jump relations as those of the Laplace or the Helmholtz equation.
- That is,

$$\lim_{\varepsilon \to 0+} S'[\sigma]\,(x \pm \varepsilon \nu_x, t) = S'[\sigma]\,(x, t) \mp \frac{1}{2}\sigma(x, t), \quad x \in \partial D.$$

$$\lim_{\varepsilon \to 0+} D[\sigma]\,(x \pm \varepsilon \nu_x, t) = D[\sigma]\,(x, t) \pm \frac{1}{2}\sigma(x, t), \quad x \in \partial D.$$

Initial-boundary value problem of the heat equation

- Consider the initial-boundary value problem of the heat equation:

$$\begin{aligned} u_t &= \Delta u + f(x,t), && (x,t) \in D \times [0,T] \\ u(x,0) &= u_0(x), && x \in D \\ u(x,t) &= g(x,t), && (x,t) \in \partial D \times [0,T]. \end{aligned} \tag{1.21}$$

- We could represent the solution as follows:

$$u(x,t) = V[f](x,t) + I[u_0](x,t) + D[\sigma](x,t). \tag{1.22}$$

  Once again, we only need to evaluate the volume and initial potentials. The only unkown here is σ, the density for the double layer potential.

- Other boundary conditions can be dealt with similarly.

Issues with the heat potentials

- Heat (layer) potentials contain integrals in both time and space.
- The direct evaluation require $O(N_T^2 N_S^2)$ work.
- The direct application of the FMM is expensive and difficult since (i). it is on $3+1$ space which increases the number of boxes in the interaction list from $6^3 - 3^3 = 189$ for 3D FMM to $6^4 - 3^4 = 1215$ for 4D FMM; (ii). the time has a preferred direction and thus needs special treatment.
- See Shaobo Wang's PhD thesis (NJIT 2016) for some progress on the evaluation of heat potentials.

Other PDEs

- Integral equation methods have also been developed for solving the Stokes equations for incompressible flows, the unsteady Stokes flow (Jiang, Veerapaneni, Greengard 2012), the Euler equations, the biharmonic equation, the modified biharmonice equation, time harmonic Maxwell's equations, and PDEs on manifolds.
- The wave equation is kind of difficult even though we know its Green's function. The reason is that it is difficult to obtain a well-condioned integral equation formulation, and it is difficult to design a stable fast algorithm for the evaluation of wave layer potentials. Currently under development.

Variable coefficient PDEs

- Consider, for example, the variable coefficient Helmholtz equation in 2D:

$$\Delta u(x) + k^2(1 + q(x))u(x) = f(x) \tag{1.23}$$

subject to the Sommerfeld radiation condition

$$\lim_{r\to\infty} \sqrt{r}\left(\frac{\partial u}{\partial r} - iku\right) = 0. \tag{1.24}$$

- One could represent u via the volume potential

$$u(x) = V[\psi](x) = \int_D G_k(x, y)\psi(y)dy,$$

where ψ is the unknown density.

- Then ψ satisfies the so-called Lippmann-Schwinger integral equation:

$$\psi(x) + k^2 q(x)V[\psi](x) = f(x).$$

That is, one needs to solve a volume integral equation instead.

Eigenvalue problems

- Integral equation methods can also be used to solve the eigenvalue problems of various PDEs including the eigenvalue problem of the Laplace problem for different domains, the transmission eigenvalue problem, the Stokes eigenvalue problems, and the mode calculation of optical waveguides (Lai and Jiang 2016).
- The basic idea is to represent the eigensolution via a suitably chosen layer potentials so that the representation satisfies the PDE in the domain and the boundary conditions lead to a set of integral equations. Here the eigenvalues usually appear as a nonlinear parameter (very often, a parameter in the fundamental solution) in the integral equations.

Others means of deriving integral equation formulation

- Besides using a linear combination and/or product of various potentials, one may also use Green's formula to derive the so-called "direct" integral equations.
- Sometimes the Dirichlet-to-Neumann map may be used.
- For the linearized Boltmann equation in nonequilibrium gas dynamics in six dimensional phase space, one may integrate along the characteristic line to obtain a system of second kind integral equations about macroscopic variables such as velocity, density, and temperature defined in physical space only.

Discretization of integral equations

- Suppose that we are trying to solve an integral equation of the form
$$A[\phi](x) = f(x), \quad x \in G$$
numerically, where $A : X \to X$ is some integral operator acting on the function space $X(G)$ with X the infinitely dimensional function space and G the underlying domain upon which the functions are defined.
- There are several methods for discretizing integral equations - projection method, collocation method, Galerkin method, Nystrom method.
- The projection method constructs a sequence of projection operators $P_n : X \to X_n$ and solve the integral equation in the finite dimensional function space X_n instead
$$P_n A \phi_n = P_n f.$$

Discretization of integral equations

- The collocation method selects a set of collocation points x_i, $i = 1, \cdots, n$ in G and try to solve

  $$A[\phi](x_i) = f(x_i), \quad i = 1, \cdots, n.$$

  We still need to discretize the involved integrals though. The collocation points are also called the supporting nodes sometimes.

- Suppose that $A : X \to Y$ with both X and Y some Hilbert spaces. Then the Galerkin method selects a finite dimensional subspace $X_n = \text{span}\{u_1, \cdots, u_n\}$ and $Y_n = \text{span}\{v_1, \cdots, v_n\}$, express the approximate solution via a linear combination of u_i, i.e., $\phi_n = \sum_{i=1}^n x_i u_i$, and tries to solve the following finite dimensional problem

  $$\sum_{i=1}^{n} (Au_i, v_j) x_i = (f, v_j), \quad j = 1, \cdots, n.$$

  Here x_i, $i = 1, \cdots, n$ are the unknown coefficients to be solved. Cons: need to evaluate double integrals!

Discretization of integral equations

- Nystrom's method. Suppose that we need to solve the following SKIE

$$\phi(x) - \int_G K(x,y)\phi(y)dy = f(x), \quad x \in G.$$

- Nystrom's method (sometimes also called the quadrature method) approximates the integral by some quadrature. That is,

$$\phi(x) - \sum_{j=1}^{n} w(x)_j K(x, y_j)\phi(y_j) = f(x), \quad x \in G.$$

  Here the quadrature weights usually depends on the target point x and thus we have denote it by $w(x)$ explicitly.

- We still need to discretize x variable. This is done via the collocation method. And we obtain the following discrete linear system:

$$\phi_i - \sum_{j=1}^{n} w_{ij} K(x_i, y_j)\phi_j = f_i, \quad i = 1, \cdots, m.$$

Discretization of integral equations

- Both the collocation method and Nystrom's method give only semi-discrete systems, and one obtains a fully discrete linear system only by combining these two methods. However, in literature, the collocation-Nystrom method is simply called Nystrom's method.
- In any case, Nystrom's method is an efficient method for solving the SKIEs. It can be viewed as a special case of the Galerkin method where one chooses the test functions as Dirac delta functions. On the other hand, most of the fully discrete implementation of Galerkin methods may be interpreted as implementations of a related collocation method.

Quadratures

- Very general remark - the currently available quadratures for smooth, weakly singular, nearly singular, singular, and hypersingular integrals depend strongly on the dimension of the integration domain.
- To summarize, there are many excellent quadratures for one dimensional integrals of almost all types. This includes classical Gaussian quadratures, generalized Gaussian quadratures, hybrid Gauss-trapezoidal rule, etc.
- For higher dimensional integrals, the number of options decreases significantly. This include tensor product quadrature, polar or spherical coordinates, Duffy quadrature, and nearly optimal quadratures for smooth functions over triangles, tetrahedra, and general convex domains in 2D by Xiao and Gimbutas, and Vioreanu and Rokhlin, and weakly singular and singular integrals over triangles by Gimbutas and Bremer.

The QBX scheme

- A. Klöckner, A. Barnett, L. Greengard, M. O'Neil, *Quadrature by expansion: A new method for the evaluation of layer potentials*, Journal of Computational Physics 252, 332-349, 2013.
- C. L. Epstein, L. Greengard, A. Klöckner, *On the convergence of local expansions of layer potentials*, SIAM Journal on Numerical Analysis 51 (5), 2660-2679, 2013.
- M. Rachh, A. Klöckner, M. O'Neil, *Fast algorithms for Quadrature by Expansion I: Globally valid expansions*, arXiv:1602.05301, 2016.
- To be discussed in detail by Andreas.

Part II. Fast Algorithms

Application background

- Two of the principal problems encountered in applied and computational mathematics are (1) the application of various linear operators (or rather, their discretizations) to more or less arbitrary vectors; and (2) solving the boundary (initial, or initial-boundary) value problems of various partial differential equations (PDE).
- Examples of linear operators: differential operator, various transformations (Legendre transform, Chebyshev transform, Laplace transform, Fourier transform, Radon transform, etc.), integral operators, etc.
- Examples of PDEs: the Laplace equation or the Poisson equation, the Helmholtz equation, the heat equation, the Schrödinger equation, the wave equation, Maxwell's equations, Navier-Stokes equations, Stokes equations, biharmonic equation, etc.

Fundamental problems

- In the end, these two problems boil down to two fundamental problems in numerical linear algebra: computing a matrix-vector product Av or solving a linear system $Ax = b$. Here the matrix A could be sparse or dense.
- Most nonlinear problems are solved numerically via some iterative scheme, for example, Newton's method or quasi-Newton's method, where a linearized problem is solved at each iteration step.
- Fast algorithms are also useful in many other situations
    - Many-body simulation in astrophysics, plasma and accelerator physics, and molecular dynamics
    - Stochastic modeling such as Browninan dynamics simulation
    - Image processing
    - Data compression in the so-called "Big Data Science".

Complexity of algorithms

- In terms of the input size, computer scientists usually divide the algorithms (or, the problems) into several groups: **P** - solvable in $\leq$ polynomial time; **NP** - solvable in $\leq$ non-deterministic polynomial time (this is a very technical definition) (for example: the game Tetris); **EXP** - solvable in $\leq$ exponential time (for example, the chess game); **R** - solvable in finite time; and problems that are not solvable in finite time. It is easy to show that most problems are actually not solvable in finite time by the current computer, which essentially relies on the facts that the set of natural number is countable, while the set of real numbers is uncountable.
- However, even **P** algorithms are simply not good enough for the purpose of scientific computing. For example, an N^3 algorithm with $N = 10^6$ is impractical for laptops.

The need of fast algorithms

- Without proper algorithms, even the most powerful supercomputer in the world is not very useful!
- **The faster the computer, the more important the speed of algorithms.** - Lloyd N. Trefethen in his essay "The Definition of Numerical Analysis" (SIAM News, 1992).
- This is because the gap between the size of the problem that can be solve by a fast (say, $O(N)$) algorithm and by a slow (say, $O(N^3)$) algorithm grows even wider as more computing resouces are available.

How fast are "fast numerical algorithms"?

- By "fast", we usually mean the algorithm has linear (i.e., $O(N)$) or quasi-linear (i.e., $O(N \log N)$ or $O(N \log^{\alpha} N)$) complexity with N the size of the input of the problem.
- Here we only use the number of floating point operations (flops) or steps to measure the computational cost (or time).
- For matrix-vector product or solving a linear system, fast algorithms automatically imply that the storage cost is no greater than $O(N)$ or $O(N \log^{\alpha} N)$ since otherwise reading the data will take more time.
- This means that we do not construct the matrix explicitly!

Changes of the computer hardware structure

- The well-known Moore's law has more or less reached its physical limit. The speed of a single CPU has not been increased significantly in the last ten years or so. Modern powerful computers are built via parallel computers, multicores, and GPUs.
- This means that the design or the re-design (i.e., the parallelization) of numerical algorithms for modern heterogeneous computers has become increasingly important. More on this in the afternoon talk by Andreas.

Very brief introduction of the following fast algorithms

- The fast multipole method (FMM) for nonoscillaroty kernels
- Fast direct solvers
- Resursive skeletonization factorization
- NUFFT - Nonuniform fast Fourier transform
- The butterfly algorithm

Key observations I - exact formulas are not always the best choice

- IEEE floating point numbers and operations are used for most problems in scientific computing and one always obtains approximate solution due to floating point arithmetic errors even if the exact formulas are used.
- For a well defined mathematical problem, the accuracy of the numerical solution depends on the condition number of the problem and the floating point arithmetic precision (double precision for almost all scientific computations nowadays).
- Our objective is to obtain a numerical solution with any prescribed accuracy, not the exact solution.
- Hence, in many cases direct methods or exact formulas are not the best choice due to high computational cost or ill-conditioning associated with them.

Key observations II - low rank property

- Many linear operators (or rather, their discretizations) in applied and computational mathematics have certain numerical low rank properties. In particular, the kernels of a wide class of integral operators becomes increasingly smooth away from the diagonal. That is, the off-diagonal blocks of the interaction matrix are of low rank regardless the size of the blocks. The FMM for nonoscillatory kernels utilizes this type of hierarchical block low-rank structure for the interaction matrix in a critical way.
- For oscillatory kernels treated by the butterfly algorithm, another type of low rank property is observed. Namely, any contiguous block of the interaction matrix having constant product of target and source box sizes has roughly the same low rank. This is the key observation of the butterfly algorithm.

The FMM for the high frequency Helmholtz kernel

- Original observation by Vladimir Rokhlin - the translation operators can be diagonalized using FFT.
- Low rank observation by Enquist and Ying in their “Fast directional multilevel algorithms for oscillatory kernels” - the interaction between a ball of radius r and a well-separated region has an approximate low rank representation, as long as the well-separated region belongs to a cone with a spanning angle of $O(1/r)$ and is at a distance which is at least $O(r^2)$ away from the ball.

Key observations III - recursive algorithmic structure

- In order to achieve optimal computational complexity, one needs to apply the low rank compression recursively.
- For fast numerical algorithms, this means one has to build a tree structure to carry out the low rank compressions in a multi-level or hierarchical way. Examples include the FMM, the butterfly algorithm, fast direct solvers, etc.
- In fact, many other fast algorithms in computer science are also recursive. Examples include quick sort, FFT, Strassen's algorithms for matrix-matrix multiplication, etc.

The original FMM - Greengard and Rokhlin, JCP 1987

Consider the Coulomb interaction in 2D: given N charges q_i at source locations $x_i \in \mathbb{C}$ $(i = 1, \cdots, N)$. Compute the potential induced by these charges at M target locations $y_j \in \mathbb{C}$ $(j = 1, \cdots, M)$.

In other words, compute

$$u_j = \sum_{i=1}^{N} q_i \log(y_j - x_i), \qquad j = 1, \cdots, M.$$

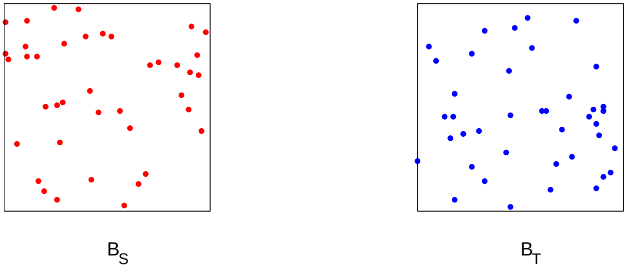
In matrix-vector notation, this is equivalent to computing the product of an M by N matrix with a column vector of length N.

The original FMM

- Obviously, direct computation costs $M \cdot N$ flops since the matrix is dense.
- The FMM reduces the computational cost down to $O(M + N)$ flops excluding the cost of sorting the particles into a tree structure. If $M = N$, then it's a reduction from $O(N^2)$ to $O(N)$.

Key low rank property for well-separated sets

Let's first consider the simplest case - the sources are well separated from targets. That is, if the source box B_S is a square of size a centered at x_0 that contains all source points x_i $(i = 1, \cdots, N)$ and the target box B_T is a square of size b centered at y_0 that contains all source points y_i $(i = 1, \cdots, M)$, then the distance between B_S and B_T is at least $\max(a, b)$.



- Using Taylor's theorem, we have

$$\log(y-x) = \log(y-x_0) - \sum_{k=1}^{\infty} \frac{1}{k} \left(\frac{x-x_0}{y-x_0} \right)^k$$

- Here $|x_i - x_0| \leq \frac{\sqrt{2}a}{2}$ and $|y_j - x_0| \geq \frac{3a}{2}$, thus $\left| \frac{x_i - x_0}{y_j - x_0} \right| \leq \frac{\sqrt{2}}{3}$ for all $i = 1, \cdots, N$ and $j = 1, \cdots, M$
- Therefore, if we truncate the series at $k = p$, then the error is at most $(\frac{\sqrt{2}}{3})^p$. And if we want the error to be less than some prescribed precision ϵ, then we should choose $p = O(\log(\frac{1}{\epsilon}))$.
- Note that p is independent of M, N and the size of the box.

Using previous result, we have

$$
\begin{aligned}
u_j &= \sum_{i=1}^{N} q_i \log(y_j - x_i) \\
&= \sum_{i=1}^{N} q_i \log(y_j - x_0) - \sum_{i=1}^{N} q_i \sum_{k=1}^{p} \frac{1}{k} \left(\frac{x_i - x_0}{y_j - x_0} \right)^k \\
&= \log(y_j - x_0) \sum_{i=1}^{N} q_i \qquad (2.25) \\
&+ \sum_{k=1}^{p} \frac{1}{(y_j - x_0)^k} \cdot (-\frac{1}{k}) \sum_{i=1}^{N} q_i (x_i - x_0)^k \\
&= \log(y_j - x_0)\Phi_0 + \sum_{k=1}^{p} \frac{1}{(y_j - x_0)^k} \cdot \Phi_k
\end{aligned}
$$

where

$$
\Phi_0 = \sum_{i=1}^{N} q_i, \qquad \Phi_k = -\frac{1}{k} \sum_{i=1}^{N} q_i (x_i - x_0)^k
$$

- This is the so-called **multipole** expansion for the potential and we may compute u_j for $j = 1, \cdots, M$ in two steps:
  Step 1: compute Φ_k for $k = 0, \cdots, p$, this costs $N \cdot p$ flops.
  Step 2: compute u_j for $j = 1, \cdots, N$, this step costs $M \cdot p$ flops.
- In matrix form, it is equivalent to a rank-p factorization of the interaction matrix $A = BC$ and then calculating Ax by $B(Cx)$.
- Altogether, the algorithm costs $(M + N) \cdot p = (M + N)\log(\frac{1}{\epsilon})$ flops.

The FMM - general adaptive nonuniform case

- In the 25 years period from 1987 to 2012, the FMM has been extended to speed up the calculation of Ax where the matrix A has the hierarchical block low-rank structure. In some literature, these are the so-called H^2 or HSS matrices or simply FMMable matrices.
- The algorithmic structure is almost identical. We first build an adaptive tree with several associated lists, then use M2M translation operator to compress the far field interactions in the upward pass, finally use M2L and L2L translation operators to obtain local expansions for evaluation in the downward pass.
- The downward pass and the translation operators are needed in order to reduce the complexity from $O(N \log N)$ to $O(N)$.

Fast direct solvers

- The fast direct solver (FDS) computes an efficient factorization of the matrix inverse A^{-1} when A is FMMable or belongs to H^2. For intrinsically 1D problems, the complexity of FDS is about linear, with the prefactor in the factorization stage ten times larger than the prefactor in the FMM, but the prefactor in the apply stage much smaller (10 or 100 times speed-up than the FMM).
- FDS are very useful when the linear system is ill-conditioned and thus requires many number of iterations when an iterative solver is used or when one needs to solve the same linear system many times with different right hand sides.
- The original FDS by Martinsson and Rokhlin (2005) recursively compresses A^{-1} using the Sherman-Morrison-Woodbury formula. So also the early work by Yu Chen on "Fast direct solver for the Lippmann-Schwinger equation" in Advances in Computational Mathematics, vol. 16, pp. 175-190, 2002.

Fast direct solvers

- For intrinsically 2D problems, FDS has $O(N^{3/2})$ complexity. And it has $O(N^2)$ complexity for 3D problems.
- The algorithmic structure of the FDS by Martinsson and Rokhlin is even more complicated than the FMM and we will not discuss it here.

Recursive skeletonization factorization

- The original FMM and FDS do not have a multiplicative factorization. Thus they are not easy to describe using the language of linear algebra and are fairly difficult to implement.
- A very recent work by V. Minden, K. L. Ho, A. Damle, and L. Ying (preprint, 2016) on "A recursive skeletonization factorization based on strong admissibility" presents a unified approach - the RS-S algorithm for both the FMM and the FDS.
- It factorizes A as a sequence of sparse and low-rank matrices and achieves linear complexity for A, A^{-1}, and $A^{1/2}$ in all dimensions as long as A has the hierarchical block low-rank property.
- As a multiplicative factorization, it is much easier to understand as we can write down the factorization explicitly in matrix form. It is also much easier to implement.

Recursive skeletonization factorization

- Three building blocks of the RS-S algorithm are block elimination via Schur complement, interpolative decomposition, and proxy surfaces.

Interpolative decomposition

Suppose that k, m, and n are positive integers with $k \leq m$ and $k \leq n$, and A is an $m \times n$ matrix, such that the rank of A is at most k. Then, there exist an $m \times k$ matrix B whose columns constitute a subset of the columns of A, and an $k \times n$ matrix P, such that

$$A_{m \times n} = B_{m \times k} P_{k \times n}$$

and

$$|P_{i,j}| \leq 1$$

for $i = 1, \cdots, k$, and $j = 1, \cdots, n$.

The decomposition of A into the product of B and P is called *interpolative decomposition* (ID) or *skeletonization* of A.

Interpolative decomposition

Here B is called the column skeleton of A. It is clear that ID is a special low rank decomposition of the matrix A. The advantage of ID (as compared with other low rank approximations) is that B is a submatrix of A and P is well-conditioned. In the setting of the FMM, this means that the multipole expansion is replaced by reassigning the charges on a subset of sources.

Similarly, ID can be performed rowwise or in both columns and rows. Algorithmically, ID is obtained by a procedure which is essentially pivoted QR factorization. ID has been used extensively in the construction of fast algorithms such as interpolative FMM, the butterfly algorithm, and fast direct solvers.

Skeletonization using ID and Schur complement

- Let $A = \begin{bmatrix} A_{pp} & A_{pq} \\ A_{qp} & A_{qq} \end{bmatrix}$ with A_{pq} and A_{qp} low rank.
- Apply ID to $\begin{bmatrix} A_{qp} \\ A_{pq}^* \end{bmatrix}$: $\begin{bmatrix} A_{qp} \\ A_{pq}^* \end{bmatrix} \approx \begin{bmatrix} A_{qr} \\ A_{rq}^* \end{bmatrix} T_p$, where T_p is an interpolation matrix with $\|T_p\|$ small.
- Reorder $A = \begin{bmatrix} A_{\tilde{p}\tilde{p}} & A_{\tilde{p}r} & A_{\tilde{p}q} \\ A_{r\tilde{p}} & A_{rr} & A_{rq} \\ A_{q\tilde{p}} & A_{qr} & A_{qq} \end{bmatrix}$, define $Q = \begin{bmatrix} I & & \\ -T_p & I & \\ & & I \end{bmatrix}$.
- Sparsify via ID: $Q_p^* A Q_p \approx \begin{bmatrix} * & * & \\ * & A_{rr} & A_{rq} \\ & A_{qr} & A_{qq} \end{bmatrix}$.
- Schur complement: $R_p^* Q_p^* A Q_p S_p \approx \begin{bmatrix} * & & \\ & * & A_{rq} \\ & A_{qr} & A_{qq} \end{bmatrix}$.

A note on Schur complement

Let

$$A = \begin{bmatrix} A_{kk} & A_{kr} & \\ A_{rk} & A_{rr} & A_{rq} \\ & A_{qr} & A_{qq} \end{bmatrix}.$$

If A_{rr} is nonsingular, define

$$R_k^* = \begin{bmatrix} I & & \\ -A_{rk}A_{rr}^{-1} & I & \\ & & I \end{bmatrix}, \qquad S_k^* = \begin{bmatrix} I & -A_{rr}^{-1}A_{kr} & \\ & I & \\ & & I \end{bmatrix}.$$

Then

$$R_k^* A S_k = \begin{bmatrix} A_{kk} & & \\ & * & A_{rq} \\ & A_{qr} & A_{qq} \end{bmatrix}$$

- Similar to LDU factorization.
- DOFs k have been eliminated, but interactions involving q are unchanged.

Proxy surfaces

- When compressing the far-field interactions, one does not need to compress the original large matrix blocks with many target points. Instead, one could replace all these target points by a fixed number of points on a so-called proxy surface and compress the interaction between the source points and these proxy points only in ID. This reduces the cost of ID to almost constant.
- Originally used in Lexing Ying's PhD thesis (2004) and the KIFMM paper by Ying, Biros, and Zorin (JCP 2004). It has become a key algorithmic ingredient in many of fast algorithms developed by Ying and others.

NUFFT - nonuniform fast Fourier transform

- The regular FFT relies on a strict algebraic structure of the discrete Fourier transform matrix and requires that points be equispaced in both physical and Fourier domains.
- The NUFFTs lift the restriction of the points being equispaced in physical and/or Fourier domain. There are now many variants of NUFFTs (see, for example, Dutt and Rokhlin 1993 and 1995, Beylkin 1995, Potts, Steidl, and Tasche 2001, Fessler and Sutton 2003, Fourmont 2003, Greengard and Lee 2004, Lee and Greengard 2005). All these NUFFTs rely on a mixture of interpolation and the judicious use of the FFT on an oversampled grid. However, Dutt and Rokhlin (1993) seems to be the first one that can achieve arbitrary precision. We will follow Greengard and Lee (2004) to explain the basic ideas and computational steps of NUFFTs.

Types of NUFFT

- Type 1 NUFFT: given the function values on an irregular grid in one space, evaluate the discrete Fourier transform on an equispaced grid in the other space. For instance,

$$F(\mathbf{k}) = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-i\mathbf{k}\cdot\mathbf{x}_j}, \tag{2.26}$$

  where one needs to evaluate $F(\mathbf{k})$ over an equispaced grid.
- Type 2 NUFFT: given the function values on an equispaced grid in one space, evaluate the discrete Fourier transform on an irregular grid in the other space.

Types of NUFFT

- For instance,

$$f(\mathbf{x}_j) = \sum_{k_1=-M/2}^{M/2-1} \sum_{k_2=-M/2}^{M/2-1} F(k_1, k_2) e^{-i(k_1,k_2)\cdot \mathbf{x}_j}, \qquad (2.27)$$

  where $\mathbf{x}_j$ are irregular grid points in the physical space.
- Type 3 NUFFT: the grid points are irregular in both the physical space and the Fourier domain. For our purpose, we only need Type 1 and 2 NUFFTs.

Basic idea and algorithmic steps of NUFFT

- For simplicity, let us consider the one dimensional type-1 NUFFT defined as follows:

$$F(k) = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{-ikx_j}, \qquad k = -\frac{M}{2}, \cdots, \frac{M}{2} - 1. \tag{2.28}$$

- The first observation is that (2.28) describes the exact Fourier coefficients of the function

$$f(x) = \sum_{j=0}^{N-1} f_j \delta(x - x_j), \tag{2.29}$$

viewed as a periodic function on $[0, 2\pi]$. Here $\delta(x)$ denotes the Dirac function. It is clearly not well-resolved by a uniform mesh in x.

Basic idea and algorithmic steps of NUFFT

- Let $g_\tau(x) = \sum_{l=-\infty}^{\infty} e^{-(x-2l\pi)^2/4\tau}$ denote the 1D periodic heat kernel on $[0, 2\pi]$. If we define $f_\tau(x) = f * g_\tau(x) = \int_0^{2\pi} f(y) g_\tau(x-y) dy$ to be the convolution of f with g_τ, then f_τ is a 2π-periodic C^∞ function and can be well-resolved by a uniform mesh in x whose spacing is determined by τ.
- Its Fourier coefficients $F_\tau(k) = \frac{1}{2\pi} \int_0^{2\pi} f_\tau(x) e^{-ikx} dx$ can be computed with high accuracy using the standard FFT on an oversampled grid.

Basic idea and algorithmic steps of NUFFT

- That is,

$$F_\tau(k) \approx \frac{1}{M_r} \sum_{m=0}^{M_r-1} f_\tau(2\pi m/M_r) e^{-ik2\pi m/M_r}, \tag{2.30}$$

  where

$$f_\tau(2\pi m/M_r) = \sum_{j=0}^{N-1} f_j g_\tau(2\pi m/M_r - x_j). \tag{2.31}$$

- Once the value $F_\tau(k)$ are known, an elementary calculation shows that

$$F(k) = \sqrt{\frac{\pi}{\tau}} e^{k^2\tau} F_\tau(k). \tag{2.32}$$

  This is a direct consequence of the convolution theorem and the fact that the Fourier transform of g_τ is $G_\tau(k) = \sqrt{2\tau} e^{-k^2\tau}$.

Some details of NUFFT

- The type-2 NUFFT can be computed simply by reversing the steps of type-1 NUFFT.
- There are some parameters in the algorithm which requires somewhat involved analysis. To summarize, we use $M_r = 2M$ and $\tau = 12/M^2$, and Gaussian spreading of each source to the nearest 24 grid points yields about 12 digits accuracy. With $\tau = 6/M^2$, Gaussian spreading of each source to the nearest 12 grid points yields about 6 digits of accuracy.

The butterfly algorithm

The butterfly algorithm speeds up the computation of

$$\phi_i(x_i) = \sum_{j=1}^{N} K(x_i, y_j) q(y_j), \qquad i = 1, \cdots, M,$$

where the kernel function K is highly oscillatory. A typical example is the discrete Fourier integral operator

$$u(x) = \sum_{k \in \Omega} e^{2\pi i \Phi(x,k)} f(k), \qquad x \in X,$$

where $\{f(k),\ k \in \Omega\}$ is a given input, $\{u(x),\ x \in X\}$ is the output, and Φ is assumed to be smooth in (x, k) for $k \neq 0$ (for example, $\Phi(x, k) = x \cdot k$. Since the kernel is highly oscillatory, it is not an HSS or H^2 matrix.

Complementary low-rank property

However, many oscillatory kernels satisfy the so-called complementary low-rank property. That is, the interaction rank between a target box A and a source box B is bounded polynomially in $\log \frac{1}{\epsilon}$ if size$(A) \cdot$ size(B) is constant. Algorithmically, suppose that A is a $N \times N$ matrix, we build a tree T_x for targets (row indices) and another tree T_n for sources (column indices) of depth $L = O(\log N)$. Then for any level l, any node A in T_x at level l, and any node B in T_Ω at level $L - l$, the interaction matrix $K_{A,B}$ has bounded rank!

Complementary low-rank property

In one dimensional cases, the condition can be restated as follows:

Any contiguous submatrix of size $m \times n$ has nearly constant low rank depending only on $m \cdot n$.

Sketch of the original butterfly algorithm

Step 1: Construct the quadtrees T_T and T_S for the target box and the source box, respectively.

Step 2: Recursive hierachical compression For $l = 1, \cdots, L$, for each pair of boxes on the level l of T_T and level $L - l$ of T_S, use ID to compress the interactions between these pairs.

Step 3: Evaluate the matrix-vector product using the compressed form of the matrix.

The algorithmic procedure is similar to that of the FFT. The algorithm is still under rapid development.

Computational cost

Two stages:

1. Precomputation stage (offline stage) - obtain an efficient factorization of the interaction matrix.
2. Apply or evaluation stage (online stage) - use the factorization to evaluate matrix-vector product.

The apply stage has $\mathrm{O}(N \log N)$ cost since usually there are $\mathrm{O}(\log N)$ factors with each factor having $O(N)$ nonzero entries.

Computational cost of the factorization stage

- $O(N^2)$ for the Helmholtz kernel by Michielssen and Borg, IEEE Trans. Antennas and Propagation, 1996.
- $O(N^2)$ for the butterfly algorithm by O'Neil and Rokhlin (2007) for various transforms of special functions - Bessel transform, Legendre transform, Fourier transform, etc.
- $O(N \log N)$ for Fourier-Integral Operators (Ying, 2009; Candes, Demanet and Ying, 2009; Demanet, Ferrara, Maxwell, Poulson, Ying, 2012, etc.).
- $O(N^{\frac{3}{2}})$ for the new butterfly factorization starting from the midlevel by Li, Yang, Martin, Ho and Ying (2015).
- $O(N \log N)$ for the so-called interpolative butterfly factorization by Y. Li and H. Yang (2016).

Software packages

- George Biros' group
  `http://padas.ices.utexas.edu/software/`
- Leslie Greengard's group
  `http://www.cims.nyu.edu/cmcl/software.html`
- Ken Ho's FLAM code
  `https://github.com/klho/FLAM`
- Andreas Klöckner
  `https://github.com/inducer`
- Lexing Ying's group
  `http://web.stanford.edu/~lexing/software/index.html`

Predictions by Lloyd N. Trefethen

Multipole methods and their descedants will be ubiquitous.
- Lloyd N. Trefethen in his essay ”Predictions for Scientific Computing Fifty Years From Now” (Mathematics Today, 2000)

Part III. The Hadamard Conjecture

History of the problem

- In 1908, Hadamard conjectured that the Green's function for the clamped plate problem, or mathematically, the first Dirichlet problem of the biharmonic equation on a convex domain is nonnegative.
- However, after 1949 numerous counterexamples disproved the positivity conjecture of Hadamard. The first result in this direction came by Duffin (1949), who showed that the Green function changes sign on a long rectangle.
- Garabedian then showed change of sign of the Green function in ellipses with ratio of half axes ≈ 1.6 (1951).
- Sign change is also proven by Coffman-Duffin (1980) in any bounded domain containing a corner, the angle of which is not too large. Their arguments are based on previous results by Osher and Seif (1973) and cover, in particular, rectangles.

The Hadamard conjecture

- This is an extremely challenging problem in analysis!
- The fundamental difficulty lies on the facts that there is no maximum principle for the biharmonic equation and that the conjecture is about the *pointwise* estimate of Green's function.
- In this workshop, we hope that we could provide some insights about positive (or sufficient) conditions for the nonnegativity of Green's function through numerical experiments, which is in perfect agreement with the objective of the ICERM (The Institute for Computational and Experimental Research in Mathematics) - probably the only mathematical institute with the word "experimental" on it.

Mathematical formulation

The Green's function for the clamped plate problem satisfies the following conditions:

$$\begin{cases} \Delta^2 G_D(x,y) = \delta(y) & \text{in} \quad D, \\ G_D = 0 & \text{on} \quad \partial D, \\ \frac{\partial G_D}{\partial n} = 0 & \text{on} \quad \partial D. \end{cases}$$

That is, it is the fundamental solution of the first kind Dirichlet problem of the biharmonic equation.

Mathematical formulation

Let us write $G_d(x, y) = u(x, y) + G(x, y)$ with $G(x, y) = \frac{1}{8\pi}|x - y|^2 \ln |x - y|$ the Green's function for the biharmonic operator in free space. Then u satisfies

$$\left\{ \begin{array}{ccc} \Delta^2 u(x, y) = 0 & \text{in} & D, \\ u = -G(x, y) := f_1(x) & \text{on} & \partial D, \\ \frac{\partial u}{\partial n_x} = -\frac{\partial G}{\partial n_x} := f_2(x) & \text{on} & \partial D. \end{array} \right.$$

SKIE formulation

- The usual single and double layer potentials do not work for this problem.
- Instead, we try to represent the solution via a sum of two multiple layer potentials

  $$u(x) = \int_S [K_1(x,y)\sigma_1(y) + K_2(x,y)\sigma_2(y)] ds_y,$$

  where K_i and σ_i $(i = 1, 2)$ are integral kernels (to be determined) and unknown densities, respectively.
- Since u has to satisfy the biharmonic equation for $x \in D$, the kernels K_i $(i = 1, 2)$ have to be a linear combination of G and its partial derivatives.

SKIE formulation

u also needs to satisfy two boundary conditions. It is well known that layer potentials may experience certain jumps across the boundary. Thus, we denote

$$\begin{aligned} K_{11}(x,y) &= \quad K_1(x,y) \\ K_{12}(x,y) &= \quad K_2(x,y) \\ K_{21}(x,y) &= \frac{\partial K_1(x,y)}{\partial n_x} \\ K_{22}(x,y) &= \frac{\partial K_2(x,y)}{\partial n_x} \end{aligned} \tag{3.33}$$

SKIE formulation

And assume that the jump relation for each associated layer potential is as follows:

$$
\begin{aligned}
&\lim_{\varepsilon\to 0+}\int_S K_{ij}(x-\varepsilon n_x, y)\sigma_j(y)ds_y = D_{ij}\sigma_j(x) \\
&+\int_S K_{ij}(x,y)\sigma_j(y)ds_y, \quad x\in S, \quad i,j=1,2,
\end{aligned}
\tag{3.34}
$$

where D_{ij} are to be determined.

SKIE formulation

With this assumption, the boundary conditions lead to the following system of integral equations in σ_i $(i = 1, 2)$

$$\begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix} \begin{pmatrix} \sigma_1(x) \\ \sigma_2(x) \end{pmatrix} +$$

$$\int_S \begin{pmatrix} K_{11}(x,y) & K_{12}(x,y) \\ K_{21}(x,y) & K_{22}(x,y) \end{pmatrix} \begin{pmatrix} \sigma_1(y) \\ \sigma_2(y) \end{pmatrix} ds_y = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}.$$

SKIE formulation

To make the above system second kind, we must require that the block diagonal matrix $D = \begin{pmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{pmatrix}$ has nonzero determinant and the integral operators K_{ij} are all compact.

SKIE formulation

The following choice of K_1 and K_2 would satify all requirements:

$$K_1(x, y) = -2G_{nnn}(x, y) + 3(\Delta G)_n(x, y) \tag{3.35}$$

$$K_2(x, y) = -G_{nn}(x, y) + G_{\tau\tau}(x, y) \tag{3.36}$$

The key here is to carry out some local singularity analysis to make sure that $D_{11} \neq 0$ and K_{11}, K_{21} are both compact.

SKIE formulation

With this selection of the kernels, we obtain the following system of second kind integral equations:

$$D(x)\sigma(x) + \int_S K(x,y)\sigma(y)ds_y = f(x), \tag{3.37}$$

where

$$D(x) = \begin{pmatrix} \frac{1}{2} & 0 \\ -\kappa(x) & \frac{1}{2} \end{pmatrix}, \quad \sigma(x) = \begin{pmatrix} \sigma_1(x) \\ \sigma_2(x) \end{pmatrix}, \quad f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}, \tag{3.38}$$

and $\kappa(x)$ is the curvatures of ∂D at x.

SKIE formulation

- This is the formulation by Peter Faskas in his PhD thesis (1989).
- It has been extended to three dimensional problems by Jiang, Ren, Tsuji, and Ying in 2011 and to modified biharmonic equation by Jiang, Kropinski, and Quaife in 2013 with slightly different choices for K_2.
- See also Manas Rachh's PhD thesis (Courant Institute, 2015) about another SKIE formulation using classical Goursat functions for 2D problems. Note that it cannot be extended to higher dimensional problems or other 4th order PDEs since it is based on complex analysis.

Numerical difficulties

- There is a severe cancellation error. The Green's function in free space is $O(1)$ as long as the target is away from the source, but the Green's function for the domain is proportional to $O(d^2)$ with d the distance from the source (or the target) to the boundary since $G_D = \frac{\partial G_D}{\partial \nu} = 0$.
- And the first sign change usually occurs when the target and source points are very close to the boundary or right on the boundary in the limiting case.
- Hence, we need high-order quadrature for singular or nearly singular integrals in order to capture the sign change accurately.
- Also, we need fast algorithms since G_D is a function of x and y. Even a linear algorithm would require $O(N^4)$ work if we need to sweep the entire domain. Here N is the total number of discretization points on the boundary. We then need to repeat this calculation for many geometries!

Possible projects

- Implement parallel FMM-accelerated QBX scheme with arbitrary precision. More on the QBX scheme in the afternoon talk by Andreas.
- Numerical experiments on ellipses to prove/disprove that Garabedian's condition is sufficient.
- Numerical experiments on arbitrary smooth convex domains with a given ratio κ_{max} to κ_{min}.
- Numerical experiments on ellipsoids in 3D as Andreas' code is dimension insensitive.