Cauchy Fast Multipole Method for Analytic Kernel

Pierre-David Létourneau[1]
Cristopher Cecka[2]
Eric Darve[1]

[1]Stanford University
[2]Harvard University

ICIAM
July 2011

Outline

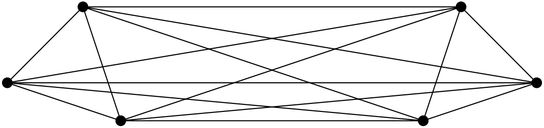
Outline

Overview

Perform matrix-vector product of the form

$$\sum_{i=1}^{N} \sigma_i K(x_i, y_j) \;\; ; \;\; j = 1..N$$

e.g. electrostatic kernel : $K(x_i, y_j) = \frac{1}{|x_i - y_j|}$



Naive implementation : $O(N^2)$

Overview

One way to speed up the computation is to *group together* particles and use a *low-rank* approximation of the kernel.

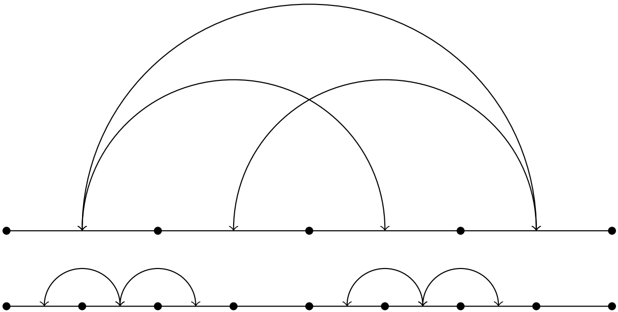
$$K(x,y) \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} a_{pq} u_p(x) v_q(y)$$

$$\sum_{i=1}^{N} \sigma_i K(x_i, y_j) \approx \sum_{q=1}^{Q} \left(\sum_{p=1}^{P} a_{pq} \left(\sum_{i=1}^{N} \sigma_i u_p(x_i) \right) \right) v_q(y_j)$$

Tree decomposition

Usually, it is not possible to get an efficient *global* separable approximation. Instead, use approximations valid only in certain *region* (far apart) and use a *tree decomposition*.

1. Use different low-rank approximation at each scale/tree level
2. Compute close-term interactions at lowest level



e.g far-neighbors (top) and close-terms (bottom)

Goal

We want to develop an algorithm with the following properties :

1. The algorithmic complexity is O(N) (up to log-factor)
2. The translation and transfer operators are *diagonal*

$$K(x,y) \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} a_{pq} u_p(x) v_q(y) \quad \text{vs} \quad \sum_{p=1}^{P} a_p u_p(x) v_p(y)$$

3. The algorithm is applicable to *any* analytic kernel

Outline

Algorithm

The algorithm is based on *Cauchy Integral formula* i.e. given $K(z)$, $z \in \mathbb{C}$, complex differentiable $\forall\, z \in \Omega$ (holomorphic), then

$$K(x) = \frac{1}{2\pi\imath} \oint_\Gamma \frac{K(z)}{z - x}\, dz \qquad (1)$$

Algorithm (Formal derivation)

Assume that $\Re(z-(x-y)) > 0 \,\forall\, z \in \Gamma$. Then, by the Laplace transform

$$K(x-y) = \frac{1}{2\pi\imath}\oint_\Gamma \frac{K(z)}{z-(x-y)}\,dz = \frac{1}{2\pi\imath}\oint_\Gamma K(z)\int_0^\infty e^{-s(z-(x-y))}ds\,dz$$

which upon discretization gives,

$$\begin{aligned} K(x-y) &\approx \frac{1}{2\pi\imath}\sum_n K(z_n)\sum_p e^{-s_p(z_n-(x-y))} \\ &= \sum_p \left(\frac{1}{2\pi\imath}\sum_n K(z_n)e^{-s_p z_n}\right) e^{s_p x} e^{-s_p y} \end{aligned}$$

This expansion possesses the desired form

Algorithm (Formal derivation)

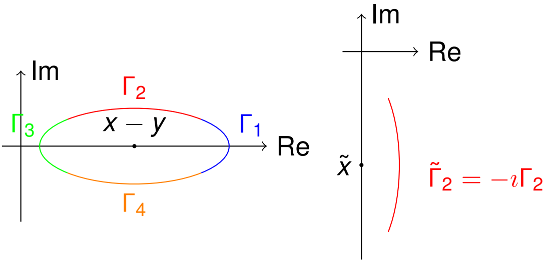
Current issues with the separable expansion :

1. It is not true that $\Re(z - (x - y)) > 0 \, \forall \, z \in \Gamma$.
2. The use of exponentials $e^{-s_k z_n}$, $e^{s_k x}$ and $e^{-s_k y}$ is numerically unstable

Both problems can however be solved.

Algorithm (Formal derivation)

To solve the first problem, we break the contour into paths and rotate each path by multiplying by $\{\lambda_k\} \in \{1, -\imath, -1, \imath\}$.



$$\frac{1}{2\pi\imath}\int_{\Gamma_k}\frac{\lambda_k}{\lambda_k}\frac{K(z)}{z-(x-y)}\,dz = \frac{\lambda_k}{2\pi\imath}\int_{\Gamma_k}K(z)\int_0^\infty e^{-s\lambda_k(z-(x-y))}\,ds\,dz$$

Algorithm (Formal derivation)

To solve the stability problem, we introduce parameters $\{\ell_k\}$ such that

$$\Re(\lambda_k z - 2\ell_k) > 0 \quad (2)$$

$$\Re(\ell_k - \lambda_k x) > 0 \quad (3)$$

$$\Re(\ell_k + \lambda_k y) > 0 \quad (4)$$

Thanks to the well-separated condition, the tree decomposition and the choice of paths, it is always possible to find such parameters. Then,

$$\int_0^\infty e^{-s\lambda_k(z-(x-y))} ds = \int_0^\infty e^{-s(\lambda_k z - 2\ell_k)} e^{-s(\ell_k - \lambda_k x)} e^{-s(\ell_k + \lambda_k y)} ds$$

which can be discretized efficiently to produce a stable approximation.

Outline

Results

Kernel studied,

- Algebraic growth (Multiquadric) : $K(x-y) = \sqrt{1+(x-y)^2}$
- Algebraic decay (Inverse Multiquadric) : $K(x-y) = \frac{1}{\sqrt{1+(x-y)^2}}$
- Oscillatory (Helmholtz) : $K(x-y) = \frac{e^{\imath k|x-y|}}{k|x-y|}$

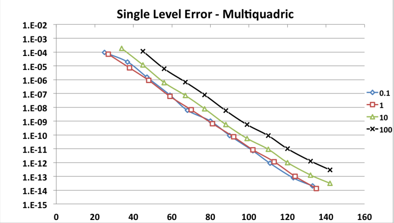
Error considered,

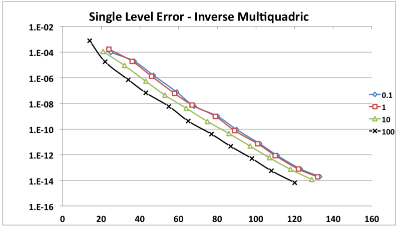
- Maximum absolute error : $\max_n |\phi_n^{FMM} - \phi_n^{Direct}|$
- Relative L^2 error : $\frac{\sqrt{\sum_n(\phi_n^{FMM} - \phi_n^{Direct})^2}}{\sqrt{\sum_n(\phi_n^{Direct})^2}}$

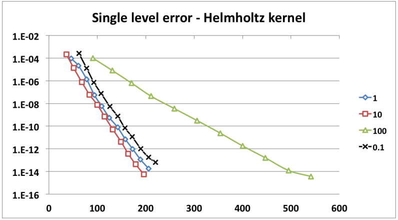
Particle distribution,

- N locations *uniformly* distributed in an interval
- Various interval size : 0.1, 1, 10, 100

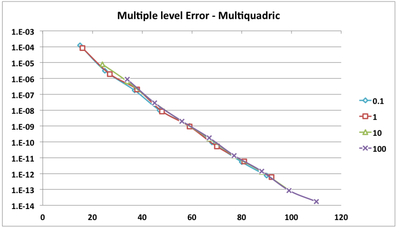
Results (Single level - Maximum absolute error vs # Quadrature points)

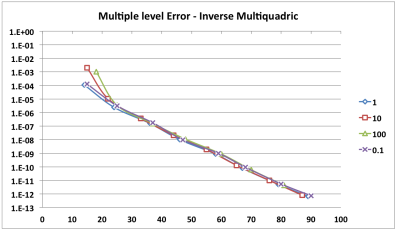


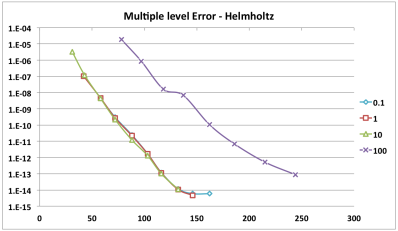




Results (Multiple levels - Relative L^2 error vs # Quadrature points)

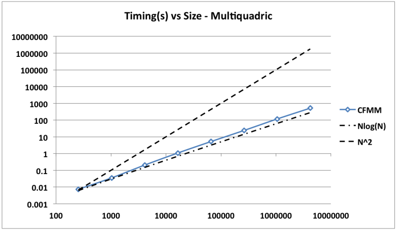


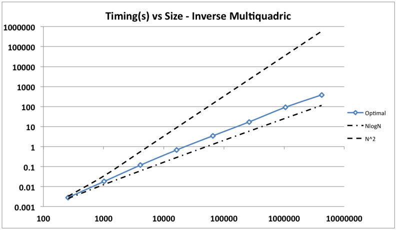


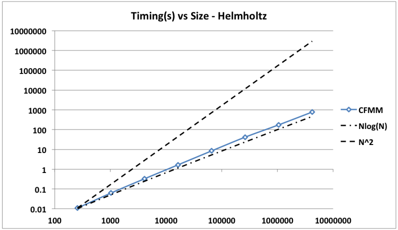


Number of particles : 65536
Number of levels : 8

Results Multiple levels - Timing vs Problem Size







Interval length : 100
Relative L^2 error : Multiquadric(10^{-6}), Inverse Multiquadric(10^{-3}), Helmholtz(10^{-5})
Computer : Dell 3GHz CPU, 8GBytes RAM

Conclusion

1. The algorithmic complexity is O(N log(N))
2. The translation and transfer operators are *diagonal*
3. The algorithm is applicable to *any* analytic kernel

4. Extension to n-D is trivial

$$K(x,y,z) = \frac{1}{(2\pi\imath)^3} \oint_{\Gamma_1} \oint_{\Gamma_2} \oint_{\Gamma_3} \frac{K(z_1,z_2,z_3)}{(z_1-x)(z_2-y)(z_3-z)} \, dz_1 \, dz_2 \, dz_3$$