Fast Multipole Methods

Pramook Khungurn

December 11, 2019

I write this document as I read notes on fast multipole methods (FMMs). The main note is "A short course on fast multipole methods" by Rick Beatson and Leslie Greengard [1].

1 Introduction

- In the N-body problem, we are given N **source points** $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$, each with associated weight $w_1, w_2, \ldots, w_N$, respectively. Then, for a **target point** $\mathbf{x}$, we wish to evaluate the sum

$$u(\mathbf{x}) = \sum_{i=1}^{N} w_i K(\mathbf{x}, \mathbf{y}_i)$$

  where the function $K(\mathbf{x}, \mathbf{y})$ is called the **kernel function**.

- For example, suppose there are N charges $q_1, q_2, \ldots, q_N$ located at points $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$. Then the electric potential at point $\mathbf{x}$ is given by:

$$\Phi(\mathbf{x}) \sum_{i=1}^{n} \frac{q_i}{\|\mathbf{x} - \mathbf{y}_i\|},$$

  and the electric field at $\mathbf{x}$ is given by:

$$\mathbf{E}(\mathbf{x}) = \sum_{i=1}^{n} q_i \frac{\mathbf{x} - \mathbf{y}_i}{\|\mathbf{x} - \mathbf{y}_i\|^3}.$$

- If there are M target points, direct evaluation of the sum for all of them would take $O(MN)$ time.

  Any algorithm that reduces the complexity to $O(Mf(N))$ where $f(N) = o(N)$ is called a **fast summation method**. The fast Fourier transform (FFT) is one example of such a method.

- Fast multipole methods can approximate the sum for M points faster than $O(NM)$. (They are referred to as "methods" instead of a "method" because they refer to a general class algorithm. One FMM is specific to one kernel.)

2 Degenerate Kernels

- If the kernel is of the form:

$$K(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{p} \phi_k(\mathbf{x}) \psi_k(\mathbf{y}),$$

  we call such a kernel **finite rank** or **degenerate**.

- In an N-body problem with such a kernel, we can precompute the **moment**:

$$A_k = \sum_{i=1}^{N} w_i \psi_k(\mathbf{y}_i)$$

for each k in $O(Np)$ time. For each target point $\mathbf{x}$, we can evaluate $u(\mathbf{x})$ in $O(p)$ time using:

$$u(\mathbf{x}) = \sum_{k=1}^{p} A_k \phi_k(\mathbf{x}).$$

- Certain FMMs rely on approximating the kernel with a degenerate kernel to gain speedups.

3 FMMs in One-Dimension

- FMMs rely on the following features:
  - a specified acceptable accuracy ϵ,
  - a hierarchical subdivision of space,
  - a far field expansion of the kernel into a degenerate kernel, and
  - the conversion of far field expansions into local expansions (optional).
- As an example, we consider the evaluation of multiquadrics radial basis function:

$$s(x) = \sum_{j=1}^{N} d_j K(x, x_j)$$

where

$$K(x, x_j) = \sqrt{(x - x_j)^2 + c^2},$$

and $0 \leq c \leq h$

- We have that, when $|x| > \sqrt{t^2 + c^2}$, the expansion

$$\begin{aligned} K(x,t) &= \sqrt{(x-t)^2 + c^2} \\ &= \text{sign}(x)\left\{ (x-t) + \frac{1}{2}c^2 x^{-1} + \frac{1}{2}tc^2 x^{-2} + \frac{1}{8}(4t^2c^2 - c^4)x^3 + \cdots + q_p(c,t)x^{-p} + \cdots \right\} \end{aligned}$$

affords the error bound

$$\left| K(x,t) - \text{sign}(x)\left\{ (x-t) + \sum_{k=1}^{p} q_k(c,t)x^{-k} \right\} \right| \leq 2(|t| + c)\left(\frac{\sqrt{t^2+c^2}}{|x|} \right)^{p+1} \frac{1}{1 - \frac{\sqrt{t^2+c^2}}{|x|}}.$$

- Notice that each term of expansion of $K(x,t)$ consists of $q_k(c,t)$, which depends on the source point alone, and x^{-k}, which depends on the target point alone. So, we have a degenerate kernel.
- Next, we subdivide the real line into contiguous intervals called **panels**. Let T denote a panel.
- For each panel, let $s_T(x)$ denote the exact contribution of all the source points in the panel:

$$s_T(x) = \sum_{j : x_j \in T} d_j K(x, x_j).$$

- Assume for the moment that the panel is centered about $x = 0$. We let $r_T(x)$ denote the far-field constribution of the panel computed by the truncated expansion of the kernel:

$$r_T(x) = \text{sign}(x)\{a_{-1}x + a_0 + a_1 x^{-1} + \cdots + a_p x^{-p}\}$$

where a_k's are the moments computed from all the points in the panel. That is:

$$a_k = \sum_{j:x_j \in T} d_j q_k(c, x_j).$$

- Suppose that $T = [-h, h)$ and $|x| \geq 3h$. We have that

$$\max_{j:x_j \in T} \frac{\sqrt{x_j^2 + c^2}}{|x|} \leq \frac{\sqrt{h^2 + h^2}}{3h} \leq \frac{\sqrt{2}}{3} = \frac{1}{2.12\ldots}.$$

Hence, the error of using $r_T(x)$ instead of $s_T(x)$ is given by:

$$|s_T(x) - r_T(x)| \leq D_T 4h \left(\frac{1}{2.12\ldots}\right)^{p+1} \frac{1}{1 - \frac{1}{2.12\ldots}} \tag{1}$$

where $D_T = \sum_{j:x_j \in T} |d_j|$.

- If the panel is $[t - h, t + h)$, which is centered at $x = t$, then the far field expansion should be centered at $x = t$. So $r_T(x)$ becomes:

$$r_T(x) = \text{sign}(x - t)\{a_{-1}(x - t) + a_0 + a_1(x - t)^{-1} + \cdots + q_p(x - t)^{-p}\}$$

where

$$a_k = \sum_{j:x_j \in T} d_j q_k(x_j - t, c).$$

The far field expansion enjoys the error bound in (1) if $|x - t| \geq 3h$.

- Points such that $r_T(x)$ converges to $s_T(x)$ at a fast rate are said to be **well separated** from T. In our case, point x is well separated from T if its distance to T is at least T's diameter. (The diameter must be at least $2c$ for this to work.)

- Now, we construct a binary tree of points by repeatedly subdividing the interval containing the points in half. Note that each node in the tree is also a panel. For each panel, we compute the coefficients a_{-1} to a_p as discussed above.

- The following function approximates $s_T(x)$ for any panel in the tree:

```
COMPUTE-SUM(x, T)
  if T is a leaf then
    return Σ_{j:x_j ∈ T} d_j K(x, x_j)
  else if x is well separated from T then
    return r_T(x)
  else
    Let R and L be the right and left children of T, respectively.
    return COMPUTE-SUM(x, R) + COMPUTE-SUM(x, L)
```

- Given ϵ, we can set $p \approx \log_{2.12}(\epsilon/(\sum_j |d_j|))$ to achieve absolute error of ϵ.

- If the tree has $O(\log N)$ level and each leaf panel has $O(1)$ source points in it, then the algorithm can approximate $s(x)$ in $O(\log N)$ time.

  The justification for this is obtained when we consider the path from the panel containing a point x to the root. At each level of the tree, we only need to do something with the node on the path and the nodes immediately to its left or to its right.

- We can make the evaluation of $s(x)$ take $O(1)$ time by noticing that, for each panel T, the expression

$$r_T = \text{sign}(x-t)(a_{-1}(x-t) + a_0 + a_1(x-t)^{-1} + \cdots + a_p(x-t)^{-p})$$

  is smooth when x is far away from t. Therefore, we can approximate it with a truncated Taylor's series expansion to get a polynomial. This polynomial gives the contribution of all source points in T to a point x far away from T.

  Now, for each panel T, we sum up the polynomials of all the panels far away from it. So, if $x \in T$, we can approximate the contribution of all points far away from T with a single polynomial evaluation. Given that there are a constant number of panels that are "near" x, the evaluation of $s(x)$ takes $O(1)$ time.

4 Fast Gauss Transform in 2D

- Given set of source points $\mathbf{x}_i = (x_i, y_i)$ with strength w_i, the function

$$U(\mathbf{x}) = \sum_{i=1}^{N} w_i e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/4T}$$

  is called the **discrete Gauss transform**.

- Define **Hermite function** $h_n(x)$ as

$$h_n(x) = (-1)^n \frac{d^n}{dx^n}(e^{-x^2}).$$

- Suppose that $\mathbf{x}_i$ is located in a square with center $\mathbf{c} = (c_1, c_2)$ and side length $\sqrt{T}$. Then,

$$e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/4T} = \sum_{n_1,n_2=0}^{\infty} \Phi_{n_1 n_2}(\mathbf{x}-\mathbf{c})\Psi_{n_1 n_2}(\mathbf{x}_i-\mathbf{c})$$

  where

$$\Psi_{n_1 n_2}(\mathbf{x}) = \frac{1}{n_1! n_2!}\left(\frac{x}{\sqrt{4T}}\right)^{n_1}\left(\frac{y}{\sqrt{4T}}\right)^{n_2}$$
$$\Phi_{n_1 n_2}(\mathbf{x}) = h_{n_1}\left(\frac{x}{\sqrt{4T}}\right)h_{n_2}\left(\frac{y}{\sqrt{4T}}\right).$$

- The error bound of the truncation of the above series expansion is:

$$\left| e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/4T} - \sum_{n_1,n_2=0}^{p} \Phi_{n_1 n_2}(\mathbf{x}-\mathbf{c})\Psi_{n_1 n_2}(\mathbf{x}_i-\mathbf{c}) \right| \leq \frac{1}{p!}\left(\frac{1}{8}\right)^p$$

  This error decays very rapidly. For four digit accuracy, choose $p = 4$. For eight digits, choose $p = 6$. For fourteen digits, choose $p = 10$.

- Suppose that all sources and targets slie in a square B_0 of unit area. Let us subdivide B_0 into squares of length $\sqrt{T}$.
- The fast Guass transform works as follows:
  1. Sort the N sourcdes into the fine squares.
  2. Choose p sufficiently large so that the error estimate is less than the desired precision ϵ.
  3. For each fine square B with center $\mathbf{c}$, compute the moments
  $$A_{n_1 n_2} = \sum_{j:\mathbf{x}_j \in B} w_j \Psi(\mathbf{x}_j).$$
  4. For target point $\mathbf{x}$, find the box B in which point $\mathbf{x}$ lies.
  Because the kernel decays exponentially fast, we ignore all but the nearest $(2n+1)^2$ boxes. This incurs the error of order $e^{-n^2/4}$, which is 10^{-4} for $n = 6$.
  The contribution of the sources in these boxes can be obtained by evaluating:
  $$\sum_{n_1, n_2 = 0}^{p} A^j_{n_1 n_2} \Phi_{n_1 n_2}(\mathbf{x} - \mathbf{c_j}).$$
  where $A^j_{n_1 n_2}$ are the precomputed moments for neighbor j whose center is $\mathbf{c}_j$.
- The running time for each query point is $O(n^2 p^2)$.

References

[1] Rick Beatson and Leslie Greengard. A short course on fast multipole methods. In *Wavelets, Multilevel Methods and Elliptic PDEs*, pages 1–37. Oxford University Press, 1997. Also available at `http://math.nyu.edu/faculty/greengar/shortcourse_fmm.pdf`.