Alcom-FT Technical Report Series
ALCOMFT-TR-03-88

Fast Relative Approximation of Potential Fields

Martin Ziegler*

University of Paderborn, 33095 GERMANY; ziegler@upb.de

Abstract. Multi-evaluation of the Coulomb potential induced by N particles is a central part of N-body simulations. In 3D, known subquadratic time algorithms return approximations up to given *absolute* precision. By combining data structures from Computational Geometry with fast polynomial arithmetic, the present work obtains approximations of prescribable *relative* error $\varepsilon > 0$ in time $\mathcal{O}(\frac{1}{\varepsilon}N \cdot \operatorname{polylog} N)$.

1 Introduction

From the very beginning a major application of computers consisted in the simulation of physical objects. Nowadays for instance so-called N*-Body Simulations* have become quite standard a tool ranging from very small particles (Molecular Dynamics) to entire galaxies (Astrophysics). Among the different kinds of attracting/repelling forces governing the motion of such point-like objects, Coulomb's (equivalently: Newton Gravitation) is both most important and most challenging: because of its slow spatial decay ('*long-range interaction*'), a fixed object experiences influences from almost any other one in the system. Naively, this leads to quadratic cost $\mathcal{O}(N^2)$ for simulating its evolution over (physical) time step by step $t \mapsto t+\tau \mapsto t+2\tau \mapsto \ldots \mapsto t+T$.

Formally, let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^3$ denote the particles' positions in physical space and $c_1, \ldots, c_N \in \mathbb{R}$ their respective charges — in case of gravitational: their masses. The respective potential and force acted by particle $\#k$ upon particle $\#\ell$ is then given (up to constants) by

$$\varphi_{k\ell} \quad = \quad c_k \cdot \frac{1}{\|\boldsymbol{x}_k - \boldsymbol{x}_\ell\|_2} \qquad \text{and} \qquad \boldsymbol{f}_{k\ell} \quad = \quad c_k \cdot \frac{\boldsymbol{x}_k - \boldsymbol{x}_\ell}{\|\boldsymbol{x}_k - \boldsymbol{x}_\ell\|_2^3} \tag{1}$$

where $\|\boldsymbol{x}\|_2 = \sqrt{\sum_i x_i^2}$ denotes Euclid's norm. Consequently, the *total* potential or force experienced by particle ℓ,

$$\Phi_\ell \quad = \quad \sum_{k \neq \ell} \varphi_{k\ell} \qquad \text{or} \qquad \boldsymbol{F}_\ell \quad = \quad \sum_{k \neq \ell} \boldsymbol{f}_{k\ell} \;, \tag{2}$$

has to be computed for each $\ell = 1, \ldots, N$ repeatedly and thus better fast. A straightforward way according to (1,2) proceeds by evaluating N sums, each ranging over $N-1$ terms: $\mathcal{O}(N^2)$. Even when exploiting symmetry to save a factor of 2, this asymptotic

* Supported by PaSCo, DFG Graduate College no.693, and by EU-Programme ALCOM-FT

severely limits scientists' desire to simulate $N \gtrsim 10^5$ particles over large scales of physical time $T \gtrsim 10^6 \tau$.

In the 2D case, a major breakthrough was achieved by GERASOULIS [6] who devised an algorithm with quasilinear cost $\mathcal{O}(N \cdot \log^2 N)$. His approach is based on fast arithmetic for complex polynomials, identifying $\mathbb{R}^2$ with $\mathbb{C}$.

In the practically more important 3D case, state-of-the-art implementations use Tree Codes and Multipole Expansions as invented by BARNES & HUT [1], taken care of worst-case distributions [4], and further improved by GREENGARD & ROKHLIN [5]. In this framework PAN & REIF & TATE [7] designed an algorithm using $\mathcal{O}(N \cdot \log N)$ many (cheap integer) operations and only $\mathcal{O}(p^2 \cdot N)$ floating point instructions to approximate the potential. They call $p = \log(C/\varepsilon)$ the "*accuracy*" of the output, where ε denotes the error bound to be achieved and $C = \sum |c_i|$ the total charge. Let us point out that C/ε, although being relative w.r.t. the total charge, does *not* necessarily describe the output precision in the relative sense; in fact, p specifies the number of terms considered in the multipole expansion for approximating the true value of the potential up to *absolute* error ε. Particularly in spatial areas of low field $\Phi_\ell \ll C$, this notion of approximation can turn out as unsatisfactory.

The present work complements [7] by approximating all Φ_ℓ up to arbitrary but fixed *relative* error $\varepsilon > 0$ within quasilinear time $\mathcal{O}(N \cdot \operatorname{polylog} N)$. A first step, Sect.2 recalls that the Euclidean norm in $\mathbb{R}^d$ permits approximation up to relative error $\varepsilon > 0$ by a certain other norm whose unit ball is a simplicial polyhedron having at most $f \leq \mathcal{O}(1/\varepsilon^{(d-1)/2})$ facets. Section 4's central Theorem 4 states that, when replacing in (1) the Euclid norm by such one, all Φ_ℓ together can be obtained exactly using only $\mathcal{O}(fN \cdot \log^{d+2} N)$ operations. Throwing things together this yields in 3D our main

Result 1 *Given $c_1, \ldots, c_N > 0$ and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^3$, one can approximate $\Phi_1, \Phi_2, \ldots, \Phi_N$ according to (2) — i.e., the value of the gravitational/Coulomb potential induced by masses/charges of respective strengths c_k at positions $\boldsymbol{x}_k$ — each up to relative error $\varepsilon > 0$, using $\mathcal{O}(\frac{1}{\varepsilon} N \cdot \log^5 N)$ operations.*

Technically speaking, we combine the Range Tree data structure from Computational Geometry [2] with fast polynomial arithmetic. Both ingredients and the way to exploit them for the above problem are discussed in Section 5 and 6, respectively.

2 Approximating the Euclidean Norm

As is well-known [9], the unit ball $B = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \leq 1\}$ of some norm $\|\cdot\|$ in $\mathbb{R}^d$ is a closed, centrally symmetric, bounded, convex set with 0 in its interior. Conversely, any such set B gives rise to a norm

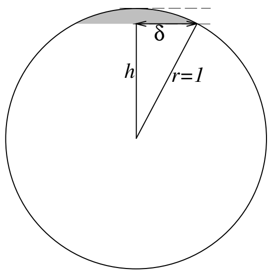
$$\mathbb{R}^d \quad \ni \quad \boldsymbol{x} \quad \mapsto \quad \inf\{\lambda > 0 : \boldsymbol{x}/\lambda \in B\}$$

having B as unit ball. We shall approximate the Euclidean norm up to relative error $\varepsilon > 0$ by replacing its smooth unit ball B with a simplicial[1] polyhedron P having 'few'

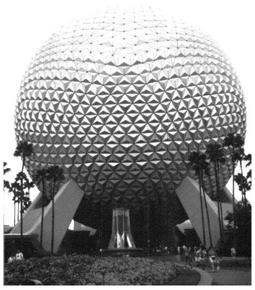
[1] each $(d-1)$-dimensional face (=*facet*) is a simplex

facets such that $(1-\varepsilon)B \subseteq P \subseteq B$. $P \cap (-P) \subseteq B$ is then a centrally symmetric, closed, bounded, and convex body containing $(1-\varepsilon)B$; it thus induces the desired norm.

Consider on the Euclidean unit ball B a spherical cap of small radius $\delta > 0$ as indicated to the right. Elementary geometry yields that the distance to the origin of any point on B's surface, after cutting off such a cap, is decreased from $r = 1$ to no less than $h = 1 - \mathcal{O}(\delta^2)$; cf. the figure below. Now recall [8] that the surface of the d-dimensional Euclidean ball can be covered by $\mathcal{O}(1/\delta)^{d-1}$ spherical caps of radius $\delta > 0$. In fact to prove this claim, ROGERS constructs a *triangulation* of B's surface of this size. We choose $\delta := \sqrt{\varepsilon}$ and take that triangulation (rather than his caps) to obtain P as above.



$h = \sqrt{1-\delta^2} \approx 1 - \delta^2/2$
for $\delta \ll 1$



EPCOT ILLUSTRATES
ROGERS' CONSTRUCTION IN 3D

3 Dealing with the 1-Norm

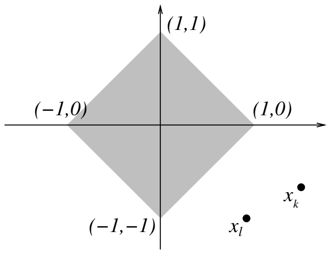
In Sect.6 we shall prove the central

Theorem 2. *Let $\psi_1, \dots, \psi_N$ denote rational functions in $z \in \mathbb{C}$, given by the coefficients of their respective nominator and denominator polynomials, all having degree at most $\Delta \in \mathbb{N}$. Upon input of these $2\Delta N$ coefficients, of $z_1, \dots, z_M \in \mathbb{C}$, and of $\boldsymbol{a}_1, \dots, \boldsymbol{a}_N \in \mathbb{R}^d$ and $\boldsymbol{b}_1, \dots, \boldsymbol{b}_M \in \mathbb{R}^d$, it is possible to jointly compute the M sums*

$$\sum_{\substack{k=1 \\ \boldsymbol{a}_k \prec \boldsymbol{b}_\ell}}^{N} \psi_k(z_\ell), \quad \ell = 1, \dots, M \qquad \textit{in time} \quad \mathcal{O}\big((M + \Delta N) \cdot \log^d(N) \cdot \log^2(\Delta N)\big)$$

where $\quad \boldsymbol{a} \prec \boldsymbol{b} \quad :\Leftrightarrow \quad \forall i = 1...d : \; a_i \prec_i b_i \qquad$ *and* $\quad \prec_i \in \{\leq, <, >, \geq\}$ *is arbitrary but fixed.*

Expressions of the form $\sum_{k:\boldsymbol{a}_k \prec \boldsymbol{b}_\ell}$ over a semi-group for given $\boldsymbol{a}$'s and one $\boldsymbol{b}_\ell$ are known in Computational Geometry as Orthogonal Range Queries. However in our case, *several* such queries are to be answered for $\ell = 1, \dots, M$; furthermore, one has to account for the preprocessing and the more complex semi-group operations involving rational functions.



To further exemplify Theorem 2, let us apply it to the computation of all $\Phi_1, \ldots, \Phi_N$ in (2) for the case of $\|\boldsymbol{x}\|_1 = \sum |x_i|$. Observe that in $\mathbb{R}^d$ this 1-norm is a simplicial polyhedral norm with 2^d facets. Moreover, restricted to some *hyper-quadrant* $\prod_{i=1}^d [0, \pm\infty)$, $\|\cdot\|_1$ is a *linear* map. In particular

$$\frac{c_k}{\|\boldsymbol{x}_k - \boldsymbol{x}_\ell\|_1} \quad = \quad \frac{c_k}{\sum_{i=1}^d x_{ki} - z}\Bigg|_{z:=\sum_{i=1}^d x_{\ell i}} \quad =: \quad \psi_k(z)$$

provided $\boldsymbol{x}_k \geq \boldsymbol{x}_\ell$ holds componentwise. Also notice that ψ_k is a rational function in $z \in \mathbb{R}$ of degree $\Delta = 1$. By letting $\boldsymbol{a}_k := \boldsymbol{x}_k$, $\boldsymbol{b}_\ell := \boldsymbol{x}_\ell$, $M := N$, one can thus compute the N sums $\sum_{k:\boldsymbol{x}_k \prec \boldsymbol{x}_\ell} \psi_{k,\ell}$ according to Theorem 2 within total time $\mathcal{O}(N \cdot \log^{d+2} N)$.

In fact by partitioning $\mathbb{R}^d \setminus \{0\}$ into disjoint (half-open/-closed) hyper-quadrants, it is possible to decompose each Φ_ℓ into 2^d many sub-sums, each one calculateable within the above time bound. In 2D for instance,

$$\Phi_\ell \quad = \quad \Phi_\ell^{(\geq,>)} + \Phi_\ell^{(<,\geq)} + \Phi_\ell^{(\leq,<)} + \Phi_\ell^{(>,\leq)}$$

where for example

$$\Phi_\ell^{(<,\geq)} \quad = \quad \sum_{\substack{k:x_{k1}<x_{\ell 1}\\ \wedge x_{k2}\geq x_{\ell 2}}} c_k \cdot \overbrace{1/\|\boldsymbol{x}_k - \boldsymbol{x}_\ell\|_1}^{=1/(x_{k2}-x_{k1}+z), \quad z:=x_{\ell 1}-x_{\ell 2}}$$

is again of the form covered by Theorem 2. Generalizing to higher dimensions in a straight-forward way, we thus have the following

Corollary 3. *A 'modified' Coulomb/gravitational potential of N particles, namely w.r.t. the 1-norm $\|\cdot\|_1$ on $\mathbb{R}^d$ instead of the Euclidean one, can be evaluated at all particles' positions exactly within only $\mathcal{O}(N \cdot \log^{d+2} N)$ operations.*

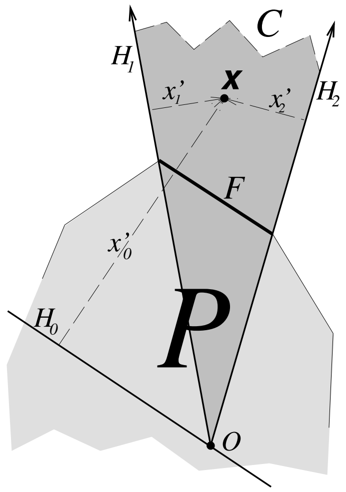
4 Dealing with Simplicial Polyhedral Norms

After this example, we now consider an arbitrary simplicial polyhedral norm $\|\cdot\|$ and its respective unit ball P. Again deal separately with each cone $C = \bigcup_{\lambda>0} \lambda F$ spanned by the origin 0 and one of P's facets F; then calculate the partial sums $\Phi_\ell^C = \sum_{k:\boldsymbol{x}_k-\boldsymbol{x}_\ell\in C} \varphi_{k,\ell}$.

In case of the 1-norm, each such C was some hyper-quadrant; fortunately the general C can be reduced to such hyper-quadrants by a simple linear transformation. To this end consider the d walls of C (simplex!) and their respective supporting hyperplanes $H_1, \ldots, H_d$, oriented such that C lies on their positive sides; let x'_i denote the signed distance of $\boldsymbol{x} \in \mathbb{R}^d$ to H_i. Linearity of $' : \mathbb{R}^d \to \mathbb{R}^d, \boldsymbol{x} \mapsto \boldsymbol{x}'$ yields:

$$\boldsymbol{x}_k - \boldsymbol{x}_\ell \in C \quad \Leftrightarrow \quad \boldsymbol{x}'_k - \boldsymbol{x}'_\ell \in [0, \infty)^d \ .$$

Finally consider H_0, the hyperplane supporting F translated to pass through the origin and oriented such that C lies on its positive side; let x'_0 denote the signed distance of $\boldsymbol{x}$ to H_0. Then for $\boldsymbol{x}_k - \boldsymbol{x}_\ell \in C$, $\|\boldsymbol{x}_k - \boldsymbol{x}_\ell\| = x'_{k0} - x'_{\ell 0}$; hence



UNIT BALL P OF A 2D POLYHEDRAL NORM

$$\sum_{k: \boldsymbol{x}_k - \boldsymbol{x}_\ell \in C} \varphi_{k,\ell} \quad = \quad \sum_{\boldsymbol{x}'_k \geq x'_\ell} \psi'_k(x'_{\ell 0}), \qquad \psi'_k(z) := c_k / (x'_{k0} - z)$$

is of the form covered by Theorem 2 and therefore can be calculated within $\mathcal{O}(N \cdot \log^{d+2} N)$ steps; indeed, each of the N linear transformations $\mathbb{R}^d \ni \boldsymbol{x}_k \mapsto (x'_{k0}, \boldsymbol{x}'_k) \in \mathbb{R}^{1+d}$ is computable with constant cost (d fixed). And that completes the proof to

Theorem 4. *Based on Theorem 2, one can evaluate a 'modified' Coulomb/ gravitational potential of N particles, namely w.r.t. a simplicial polyhedral norm $\|\cdot\|$ on $\mathbb{R}^d$ instead of the Euclidean one, exactly within $\mathcal{O}(fN \cdot \log^{d+2} N)$ operations where f denotes the number of facets of $P := \{x \in \mathbb{R}^d : \|x\| \leq 1\}$.*

It thus remains to prove Theorem 2. To this end, we shall construct and exploit a Range Tree data structure as is known from Computational Geometry. However for our purpose, nodes correspond to (ranges of) rational functions rather than to (ranges of) points; similarly, a query does not report points but performs multi-evaluations. The costs inferred by the latter and some other operations on rational functions are recalled in the next section before proceeding in Sect.6 to the actual proof of Theorem 2.

5 Fast Arithmetic for Rational Functions

An important ingredient to our algorithm is fast arithmetic for polynomials in one complex variable z, based on Fast Fourier Transformation (FFT). For later reference, we briefly collect the running times for two basic operations.

Lemma 5. *a) The product of two polynomials of degree at most n, both input and output represented by respective lists of coefficients, can be computed in time $\mathcal{O}(n \cdot \log n)$.* [3, THEOREM 2.8]

b) *A polynomial of degree at most n can be multi-evaluated at m given points simultaneously within total time $\mathcal{O}\big((m+n)\cdot\log^2 n\big)$.* [3, COROLLARY 3.20]

It is worth while emphasizing that the above upper complexity bounds, although quoted from a theory book, are well-known in practice and correspond to highly efficient algorithms.

Lemma 5 obviously holds as well for rational functions in one complex variable z represented in the form *nominator/denominator*, both being polynomials of maximal degree n and given by their respective coefficient lists. To be precise when talking about arithmetic on rational functions, let us agree the value of $\psi = \alpha/\beta$ with $\alpha, \beta \in \mathbb{C}[Z]$ at some zero z of β to be undefined $\psi(z) := \infty$ even in case of a *removable* singularity, i.e., even when $\alpha(z) = 0$ is a zero of same or higher order. Furthermore $z \pm \infty := \infty$, $z \cdot \infty := \infty$, and $z/\infty := \infty$ for all $z \in \mathbb{C} \cup \{\infty\}$.

Corollary 6. *a) The product of two rational functions of degree at most n can be computed in time $\mathcal{O}(n \cdot \log n)$.*

b) *A rational function of degree at most n can be evaluated at m given points simultaneously within total time $\mathcal{O}\big((m+n)\cdot\log^2 n\big)$.*

c) *The sum of two rational functions of degree at most n can be computed in time $\mathcal{O}(n \cdot \log n)$.*

The last claim holds by calculating the sum of two rational functions according to

$$\psi_1 + \psi_2 = \frac{\alpha_1(z)}{\beta_1(z)} + \frac{\alpha_2(z)}{\beta_2(z)} = \frac{\alpha_1(z)\cdot\beta_2(z) + \alpha_2(z)\cdot\beta_1(z)}{\beta_1(z)\cdot\beta_2(z)}$$

using three polynomial multiplications and two additions. The result is then of degree no more than $\deg(\psi_1) + \deg(\psi_2)$.

6 Interval Tree of Rational Functions

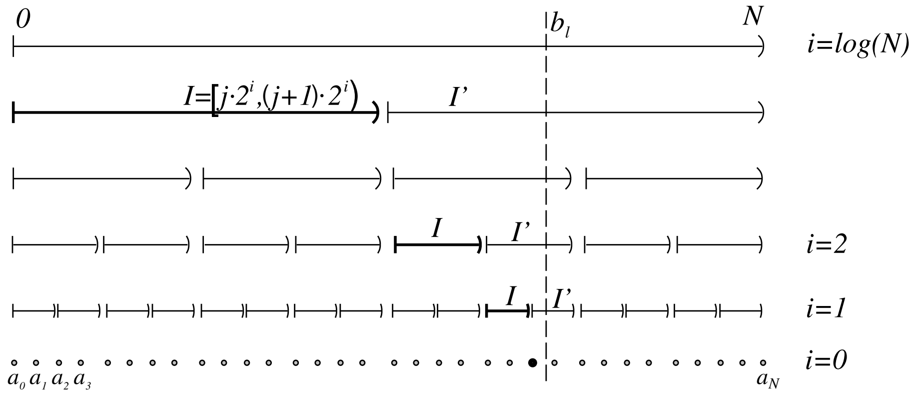
We finally come to prove the announced

Theorem 2. *Let $\Psi = \{\psi_1, \ldots, \psi_N\}$ denote a family of rational functions in $z \in \mathbb{C}$ of degree at most $\Delta \in \mathbb{N}$, given by their respective coefficients. Upon further input of a family $Z = \{z_0, \ldots, z_{M-1}\} \subset \mathbb{C}$, of $A = \{\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{N-1}\} \subset \mathbb{R}^d$, and of $B = \{\boldsymbol{b}_0, \ldots, \boldsymbol{b}_{M-1}\} \subset \mathbb{R}^d$, it is possible to jointly compute the M sums*

$$\sum_{k:\boldsymbol{a}_k \prec \boldsymbol{b}_\ell} \psi_k(z_\ell), \qquad \ell = 0, ..., M-1$$

in time $\mathcal{O}\big((M + \Delta N)\cdot\log^d(N)\cdot\log^2(\Delta N)\big)$.

The proof proceeds by induction on d, starting with dimension 1. W.l.o.g. let N a power of 2, $a_0 < a_1 < \ldots < a_{N-1} < a_N := \infty$, $'\prec' = '<'$.



EXAMPLE OF AN INTERVAL TREE, FOR SIMPLICITY ON DATA $a_k \equiv k$.

The Interval Tree is a complete binary tree with N leafs; each level $i = 0, 1, \ldots$, $\log(N)$ corresponds to a partition of the real interval $[a_0, \infty) \subseteq \mathbb{R}$ into $N/2^i$ subintervals $I = \big[a_{j\cdot 2^i}, a_{(j+1)\cdot 2^i}\big)$, $j = 0, \ldots, N/2^i - 1$. For each such $I = I_{i,j}$, $j < N/2^i - 1$, let $I' := I_{i,j+1}$ denote its right neighbor.

The algorithm performs the following steps:

A) Construct the above tree.
B) For each $i = 0, \ldots, \log(N)$
C) For each $\ell = 0, \ldots, M-1$
determine the unique I_{ij} containing b_ℓ, $j = 0, \ldots, N/2^i - 1$;
endfor ℓ; this gives rise to a partition $Z_{ij} := \{z_\ell : b_\ell \in I_{ij}\}$ of Z.
D) For each $j = 0, \ldots, N/2^i - 1$
i) compute (the coefficients of) $\psi_I := \sum_{k:a_k \in I} \psi_k, \quad I = I_{ij}$;
ii) multi-evaluate ψ_I on those z_ℓ with $b_\ell \in I'$ as obtained in C)
endfor j
endfor i.
E) For each $\ell = 0, \ldots, M-1$
compose the desired sum $\sum_{k:a_k < b_\ell} \psi_k(z_\ell)$ by adding up at most $\log(N)$
many of the precomputed $\psi_I(z_\ell)$.

Looking at the above sketch of an Interval Tree, one easily confirms that the latter composition is always feasible. Concerning the analysis, Step E) obviously contributes $\mathcal{O}(N \cdot \log N)$ to the total running time. Step A) costs $\mathcal{O}(N \cdot \log N)$. Within the j-loop, Step C) can be performed in time $\mathcal{O}\big(M \cdot \log(N/2^i)\big)$ by means of binary search. Step i) of the i-loop costs $\mathcal{O}\big(\Delta 2^i \cdot \log(\Delta N)\big)$ according to Corollary 6d); indeed, ψ_I is a rational function of degree at most $\Delta 2^i$ and furthermore the sum of two others, say ψ_J and $\psi_{J'}$, the already have been computed during pass $i-1$. Step ii) finally takes $\mathcal{O}\big((m_{ij} + \Delta 2^i) \cdot \log^2(\Delta N)\big)$ according to Corollary 6b), $m_{ij} := |Z_{ij}|$.

Since the Z_{ij} partition Z, $\sum_j m_{ij} \leq M$; one single pass of the i-loop costs

$$\mathcal{O}\big(M \cdot \log(N/2^i)\big) \quad + \quad \mathcal{O}\big(\Delta N \cdot \log(\Delta N)\big) \quad + \quad \mathcal{O}\big((M + \Delta N) \cdot \log^2(\Delta N)\big)$$

which is $\leq \mathcal{O}\big((M + \Delta N) \cdot \log^2(\Delta N)\big)$. Counter i running from 0 to $\log(N)$ infers another factor of $\log(N)$, and that completes the induction start. $\square$

Observe that already the just proved case $d=1$ allows to reproduce GERASOULIS' result [6] on generalized Hilbert Matrices (although with one additional $\log$-factor):

Corollary 7. *Given $c_1, \ldots, c_N \in \mathbb{C}$ and pairwise distinct $z_1, \ldots, z_N \in \mathbb{C}$, one can compute the N sums* $\displaystyle\sum_{k \neq \ell} \frac{c_k}{z_k - z_\ell}, \quad \ell = 1, \ldots, N$ *within time $\mathcal{O}(N \cdot \log^3 N)$.*

Indeed, $\psi_k(z) := \frac{c_k}{z_k - z}$ is a rational function of degree $\Delta = 1$; hence according to Theorem 2, both parts of $\sum_{k\neq\ell} \psi_k(z_\ell) = \sum_{k<\ell} \psi_k(z_\ell) + \sum_{k>\ell} \psi_k(z_\ell)$ can be obtained for $\ell = 1, \ldots, N$ within the claimed running time by choosing $a_k := k \in \mathbb{R}^{\mathbf{1}}$ and $b_\ell := \ell \in \mathbb{R}^{\mathbf{1}}$.

7 Range Tree of Rational Functions

The induction step $d \mapsto d+1$ in our proof for Theorem 2 borrows Range Trees from Computational Geometry [2]. For data $(\alpha_k, \boldsymbol{a}_k) \in \mathbb{R}^{1+d}$, this is defined recursively to be a (1D) Interval Tree on data $\alpha_k \in \mathbb{R}$ with nodes consisting of d-dimensional Range Trees on (appropriate ranges of) data $\boldsymbol{a}_k \in \mathbb{R}^d$.

In our case, we use it to store, process, and multi-evaluate sums of rational functions. For $(\beta_\ell, \boldsymbol{b}_\ell) \in \mathbb{R}^{1+d}$, $\ell = 0, \ldots, M-1$, the algorithm proceeds as follows:

A) Construct an Interval Tree on data α_k, $k = 0, \ldots, N-1$.
B) For each $i = 0, \ldots, \log(N)$
C) For each $j = 0, \ldots, N/2^i - 1$
 i) determine $A_{ij} := \{\boldsymbol{a}_k : \alpha_k \in I_{ij}\}$ and $\Psi_{ij} := \{\psi_k : \alpha_k \in I_{ij}\}$;
 ii) determine $B_{ij} := \{\boldsymbol{b}_\ell : \beta_\ell \in I_{ij}\}$ and $Z_{ij} := \{z_\ell : \beta_\ell \in I_{ij}\}$;
 iii) apply Theorem 2 to these *sub*collections $\Psi_{i,j} \subseteq \Psi$, $Z_{i,j+1} \subseteq Z$, $A_{i,j} \subseteq A$, and $B_{i,j+1} \subseteq B$.
 This will yield all sums $\displaystyle\sum_{\substack{k:\alpha_k \in I\\ \boldsymbol{a}_k < \boldsymbol{b}_\ell}} \psi_k(z_\ell)$ for $\beta_\ell \in I'$.
 endfor j
 endfor i.
D) For each $\ell = 0, \ldots, M-1$
 compose the desired sum $\displaystyle\sum_{\substack{k:\alpha_k < \beta_\ell\\ \boldsymbol{a}_k < \boldsymbol{b}_\ell}} \psi_k(z_\ell)$ by adding up at most $\log(N)$
 many of the precomputed $\displaystyle\sum_{\substack{k:\alpha_k \in I\\ \boldsymbol{a}_k < \boldsymbol{b}_\ell}} \psi_k(z_\ell)$ with $\beta_\ell \in I'$.

Again, the latter composition is always feasible due to the Interval Tree's properties. Like in Section 6, the running times are dominated by Step iii) of the innermost loop. Let

$m_{ij} = |Z_{ij}| = |B_{ij}|$; observe $n := |\Psi_{ij}| = |A_{ij}| = 2^i \le N$ and again $\sum_j m_{ij} \le M$. By induction hypothesis, a whole run of the j-loop thus infers cost

$$\mathcal{O}\Big(\sum_{j=0}^{N/2^i-1} \big((m_{ij} + \Delta n) \cdot \log^d(n) \cdot \log^2(\Delta n) \big) \Big) \\ \le \quad \mathcal{O}\big((M + \Delta N) \cdot \log^d(N) \cdot \log^2(\Delta N) \big)$$

and the i-loop gives another factor $\log(N)$. □

8 Conclusion

We presented a quasilinear time algorithm for a central problem of N-body simulations: determining for each particle the potential it experiences by the other particles. Other than previous approaches, our approximation achieves guaranteed *relative* errors. Whereas [7] used a synthesis of Algebraic and Numerical Computing, we combine Algebraic Computing with Computational Geometry. This technique yields (Theorem 4) that for a simplicial polyhedral norm with ’few’ facets rather than the Euclidean one, the potential fields can be evaluated *exactly* in quasilinear time. Since the Euclid norm permits approximation by simplicial polyhedral norms with ’few’ facets (Sect.2), the claim follows.

In fact, Theorem 4 immediately generalizes to higher integral (positive or negative) *powers* of simplicial polyhedral norms:

Theorem 8. *Fix $d \in \mathbb{N}$, $q \in \mathbb{Z}$, and let $\|\cdot\|$ denote a simplicial polyhedral norm in $\mathbb{R}^d$ with f facets. Then one can compute, upon input of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathbb{R}^d$ and $c_1, \dots, c_N \in \mathbb{R}$, all*

$$\Phi_\ell \quad = \quad \sum_{k \ne \ell} c_k \cdot \|\boldsymbol{x}_k - \boldsymbol{x}_\ell\|^q, \qquad \ell = 1, \dots, N$$

exactly using $\mathcal{O}\big(f|q|N \cdot \log^d(N) \cdot \log^2(qN)\big)$ operations.

To be fair, Result 1 requires *positive* strengths $c_k > 0$. In case of gravitation that condition is naturally met by all masses; whereas Coulomb charges (ions) do occur with different signs. In practice one would simply treat the positive and negative ones separately and then subtract their respective contributions. However in strict terms of worst-case analysis, doing so violates the relative error bounds: the difference of two approximations may lead to infinite relative deviations.

Another issue, the high powers of $\log(N)$ in our results might already in 3D not be as negligible as the common notion ”*quasilinear time*” suggests. Of course, $\mathcal{O}(N \cdot \operatorname{polylog} N)$ does pay off eventually against the naive $\mathcal{O}(N^2)$ approach as $N \to \infty$; *when* break-even occurs is about to be examined in practical implementations. Theoretically, Fractional Cascading [2] might help removing at least one factor $\log(N)$. Also it is quite conceivable that multi-evaluation (Lemma 5b) has only complexity $\mathcal{O}\big((m+n) \cdot \log^{\mathbf{1}} n\big)$, thus saving another $\log(N)$.

However the most central open question is of course this: *Can 3D Coulomb fields be evaluated exactly in subquadratic time?*

References

1. BARNES, J., AND P. HUT: "A hierarchical $\mathcal{O}(N \cdot \log N)$ force-calculation algorithm", pp.446-449 in *Nature* Vol.**324** (1986).
2. DE BERG, M., AND M. VAN KREVELD, AND M. OVERMARS, AND O. CHEONG: "Computational Geometry", Springer (2000).
3. BÜRGISSER, P., AND M. CLAUSEN, AND M.A. SHOKROLLAHI: "Algebraic Complexity Theory", Springer (1997).
4. CALLAHAN, P.B., AND S.R. KOSARAJU: "A Decomposition of Multidimensional Point Sets with Application to k-Nearest-Neighbors and n-Body Potential Fields", pp.67-90 in *J. ACM* vol.**42** (1995); see also
   ANDERSON, R.J.: "Tree Data Structures for N-Body Simulation", pp.1923-1940 in *SIAM Journal on Computing* vol.**28**(**6**) (1999).
5. CHENG, H., AND L. GREENGARD, AND V. ROKHLIN: "A Fast Adaptive Multipole Algorithm in Three Dimensions", pp.468-498 in *Journal of Computational Physics* Vol.**155** (1999).
6. GERASOULIS, A.: "A Fast Algorithm for the Multiplication of Generalized Hilbert Matrices with Vectors", pp.179-188 in *Mathematics of Computation* Vol.**50** No.**181** (1988).
7. PAN, V., AND J.H. REIF, AND S.R. TATE: "The Power of Combining the Techniques of Algebraic and Numerical Computing: Improved Approximate Multipoint Polynomial Evaluation and Improved Multipole Algorithms", pp.703-713 in *Proc. 32th Annual IEEE Symposium on Foundations of Computer Science* (FOCS'1992); see also
   REIF, J.H., AND S.R. TATE: "N-body Simulation I: Fast Algorithms for Potential Field Evaluation and Trummer's Problem", *Tech. Report* No.**N-96-002**, Univ. of North Texas Dept. of Computer Science (1996).
8. ROGERS, C.A.: "Covering a Sphere with Spheres", pp.157-164 in *Mathematika* Vol.**10** (1963).
9. RUDIN, W.: "Functional Analysis", McGraw-Hill (1991).