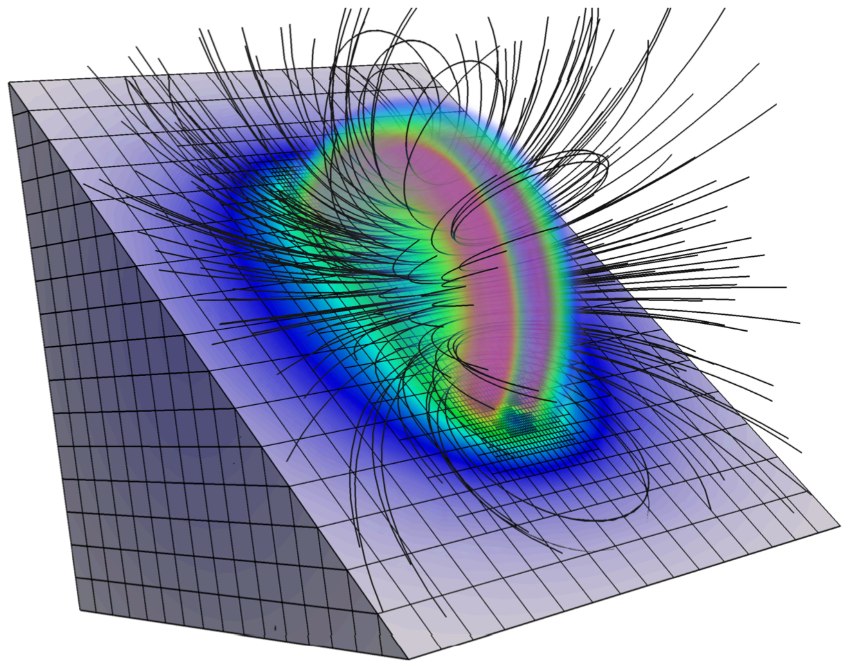
# N-body methods in Computational Science and engineering

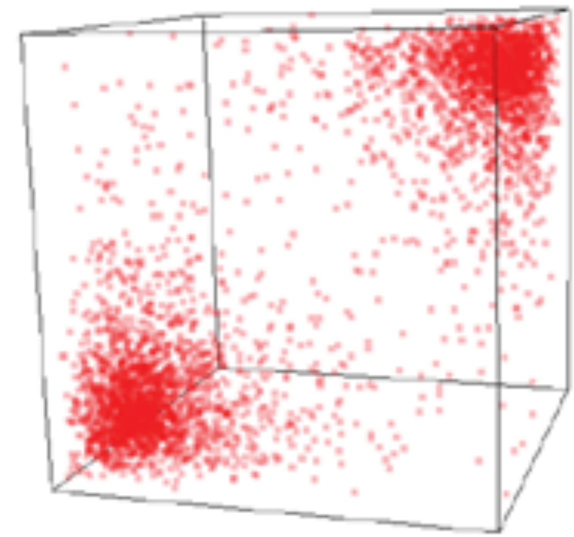
I. Intro to N-body
II. Algebraic View



George Biros

INSTITUTE FOR COMPUTATIONAL ENGINEERING & SCIENCES
THE UNIVERSITY OF TEXAS AT AUSTIN

# N-body problem



**Input**

$N$ points in $\mathbb{R}^d$: $x_1, \ldots, x_N$

$N$ densities in $\mathbb{R}$ : $w_1, \ldots, w_N$

**Output**

$N$ potentials in $\mathbb{R}$ : $u_1, \ldots, u_N$

$$u_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} G(x_i, x_j) w_j$$

$$G(x_i, x_j) = \frac{1}{\|x_i - x_j\|_2}$$

Related to convolutions
$u(x) = \int G(x, x') w(x')$

# Fast N-body solver milestones (up to 2000)

Vladimir Rokhlin, Yale

Leslie Greengard, Courant Institute

## Sequential

Appel, 1985 - O(N logN)
Barnes & Hut, 1986 - O(N logN) tree code
Rokhlin, 1985 - Multipole translations
**Greengard & Rokhlin, 1987** - FMM O(N)
Carrier, Greengard, & Rokhlin - Adaptive FMM O(N)
Anderson, 1992 - Equivalent densities
Cheng, Greengard & Rokhlin, 1999 - Efficient 3D FMM

## Parallel

Greengard & Gropp, 1991 - Uniform distributions
Warren & Salmon, 1992 - Distributed treecode
SH Teng, 1998 - Theory

## Different Operators

Greengard & Strain, 1991 – Gauss transform
Rohklin, 1992 – High frequency Helmholtz
Michielssen, 1998 – Time-domain Wave, Maxwell
Kapur & Long, 1997 – Algebraic view

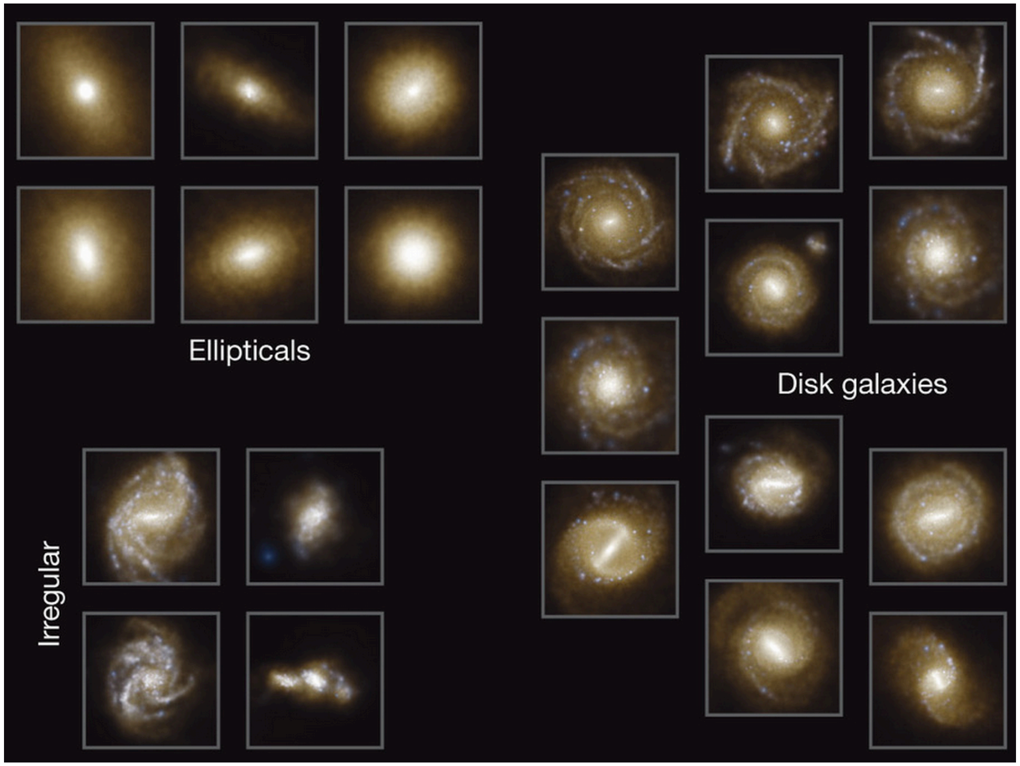
# App – gravitational interactions

## Properties of galaxies reproduced by a hydrodynamic simulation

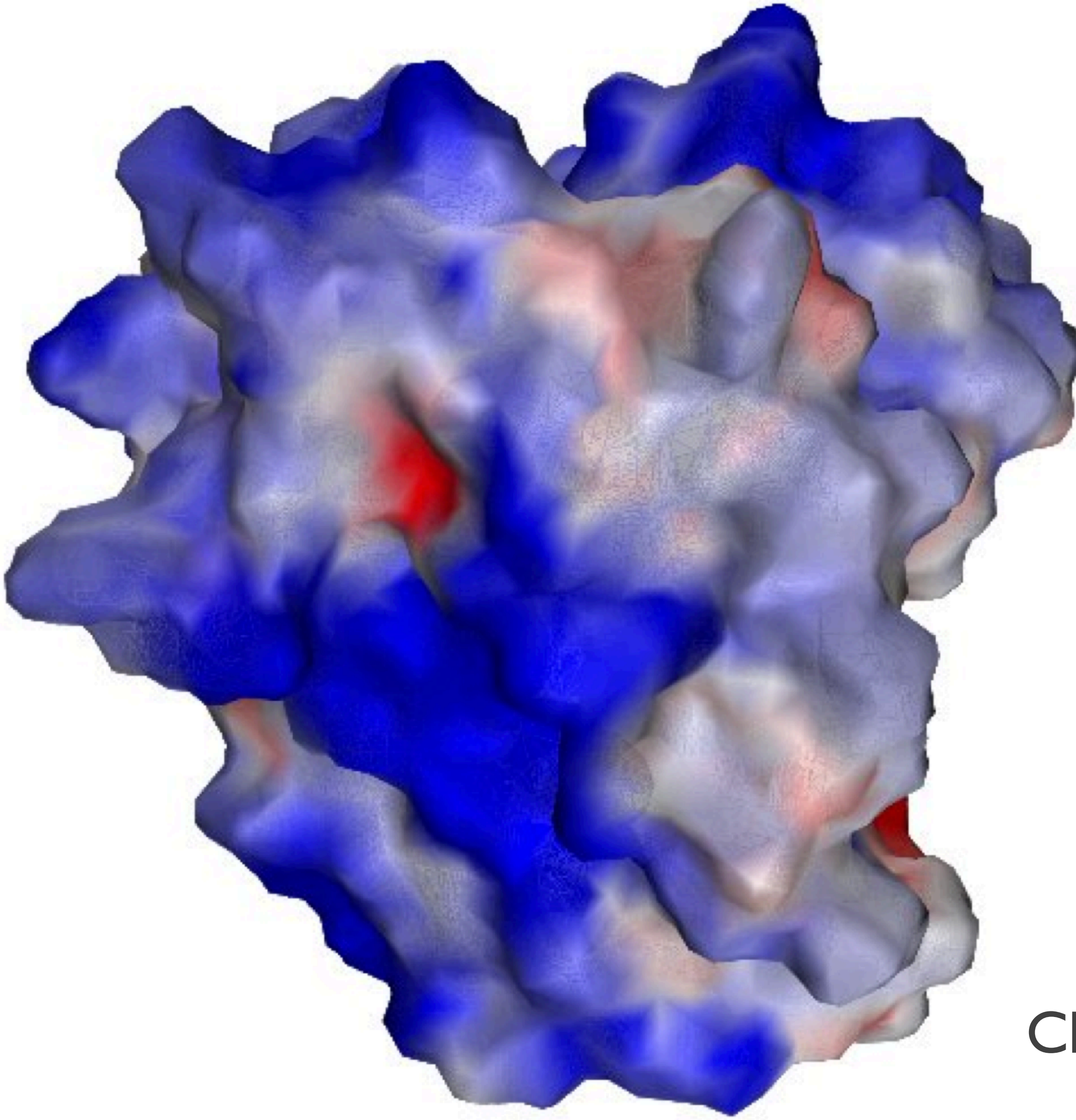
M. Vogelsberger, S. Genel, V. Springel, P. Torrey, D. Sijacki, D. Xu, G. Snyder, S. Bird, D. Nelson & L. Hernquist

Affiliations | Contributions | Corresponding author

*Nature* **509**, 177–182 (08 May 2014) | doi:10.1038/nature13316
Received 27 January 2014 | Accepted 07 April 2014 | Published online 07 May 2014

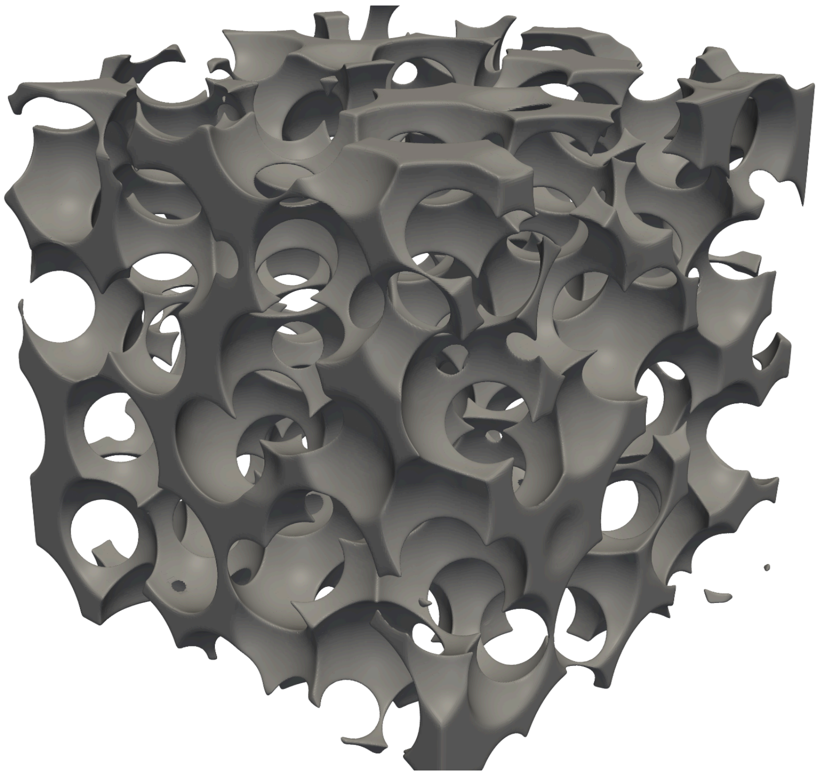


# App – protein electrostatics



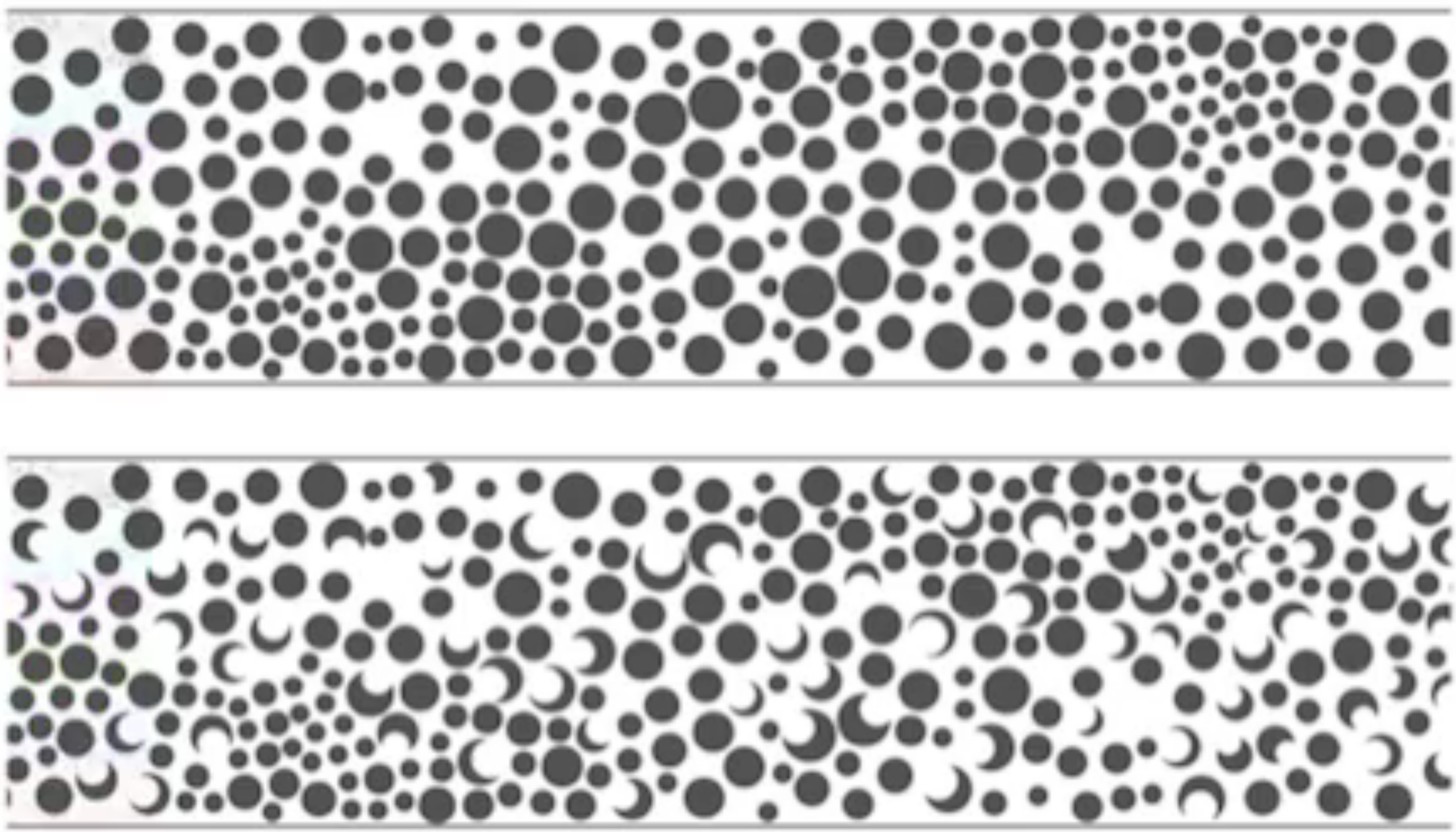
Chandra Bajaj, UT Austin

# App – flows in porous media with VIEs



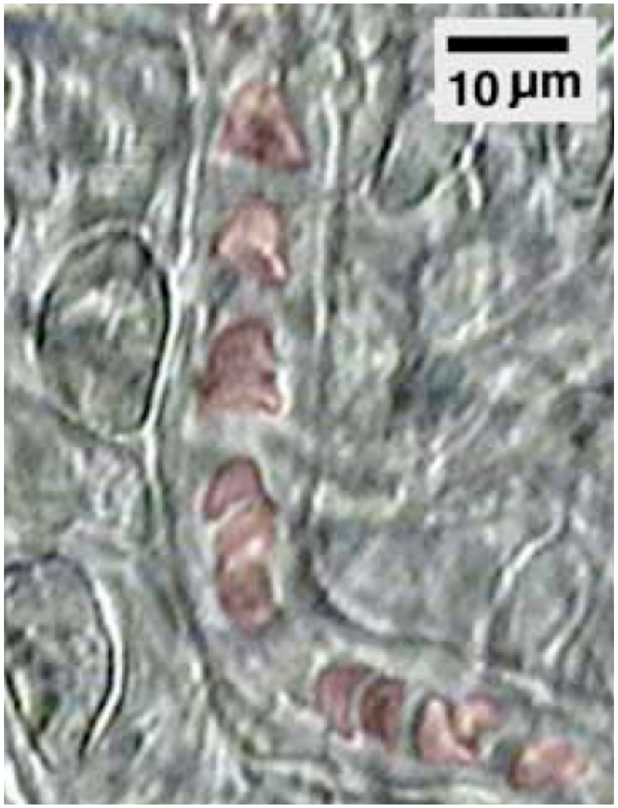
Malhotra, Gholami, B.  SC14

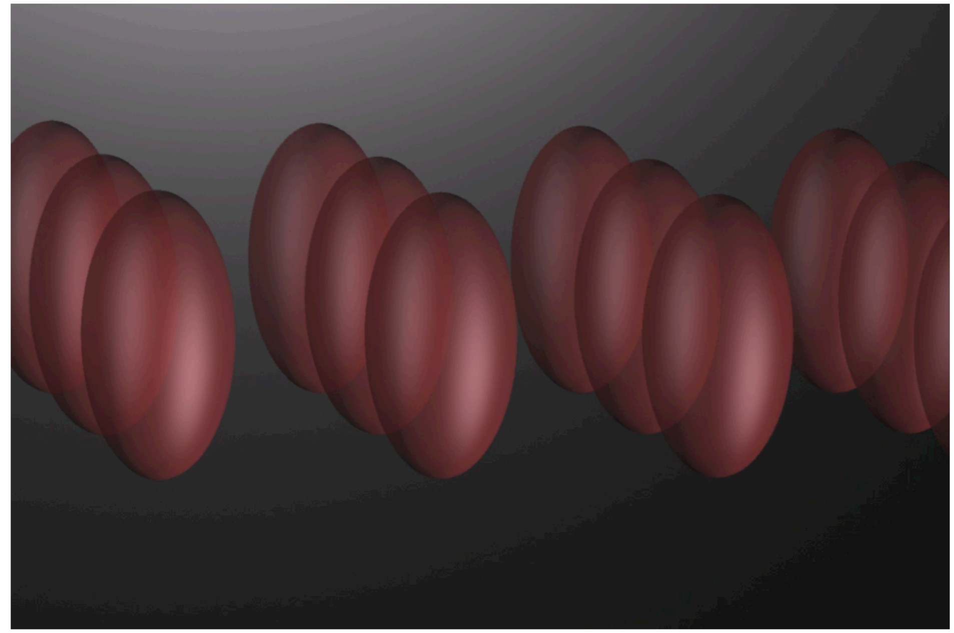
# App-Porous media with BIEs

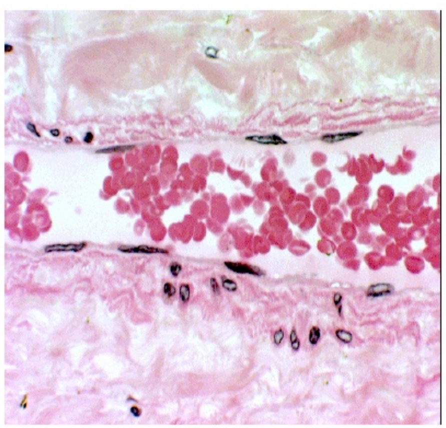


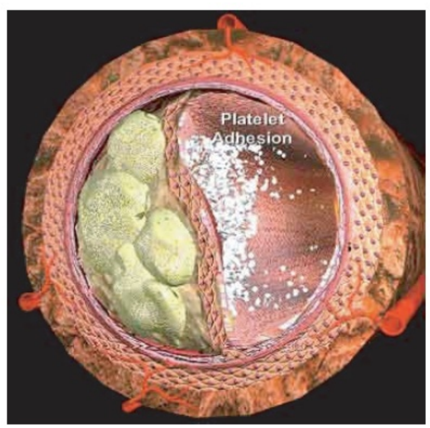
with B. Quaife (UT Austin), P. de Anna (MIT), R. Juanes (MIT)

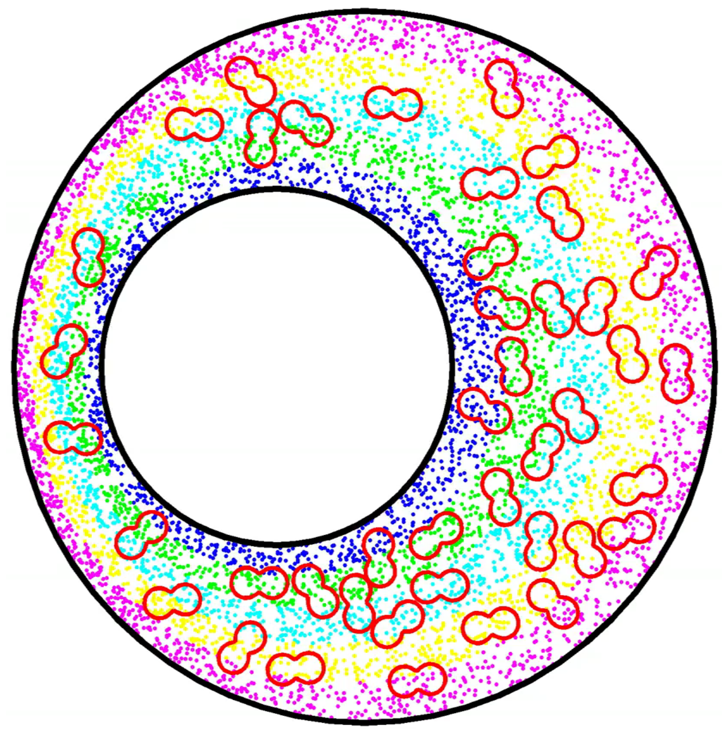
# App-Complex fluids



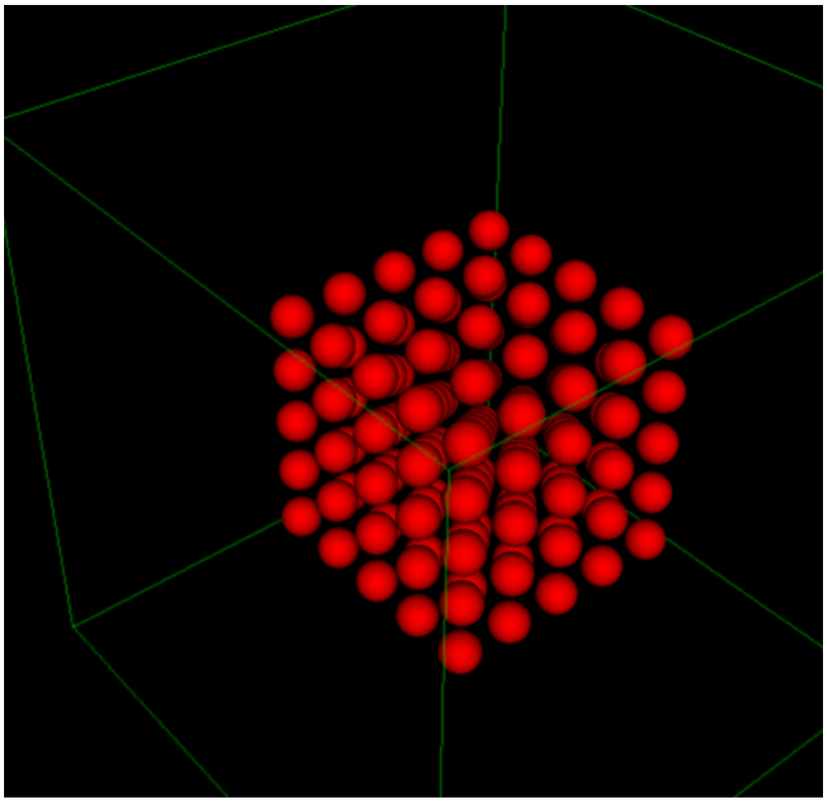



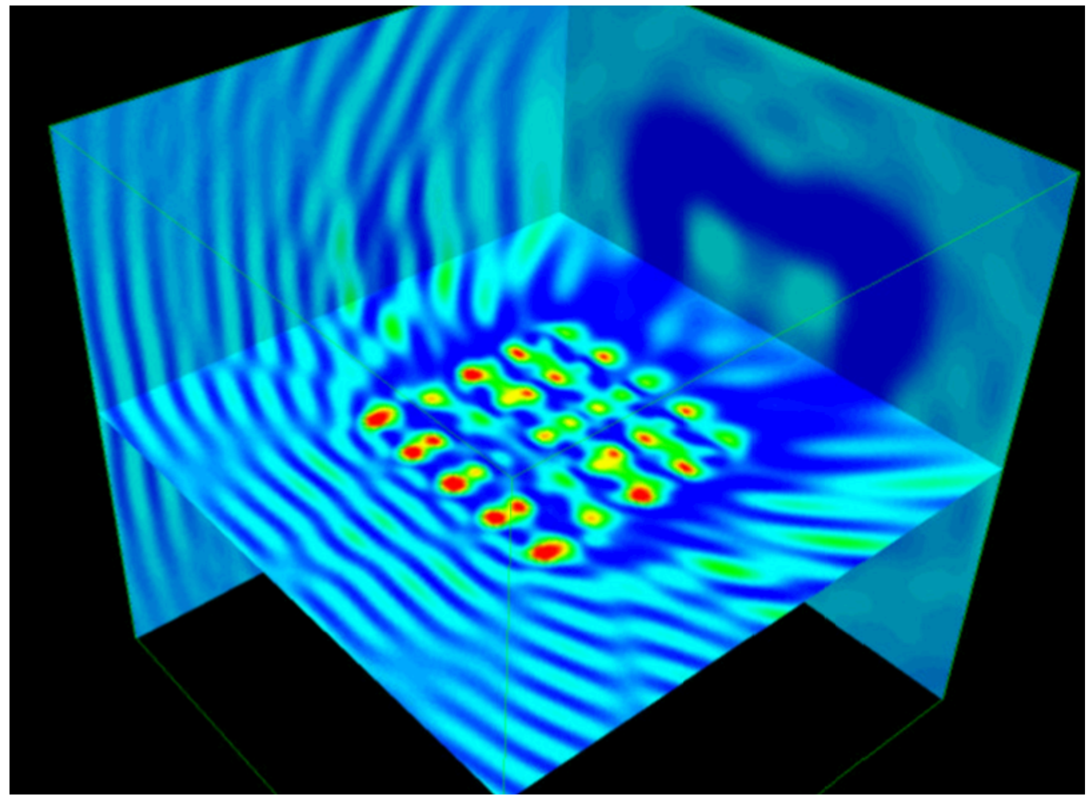






# Electromagnetic scattering (time-harmonic Maxwell)



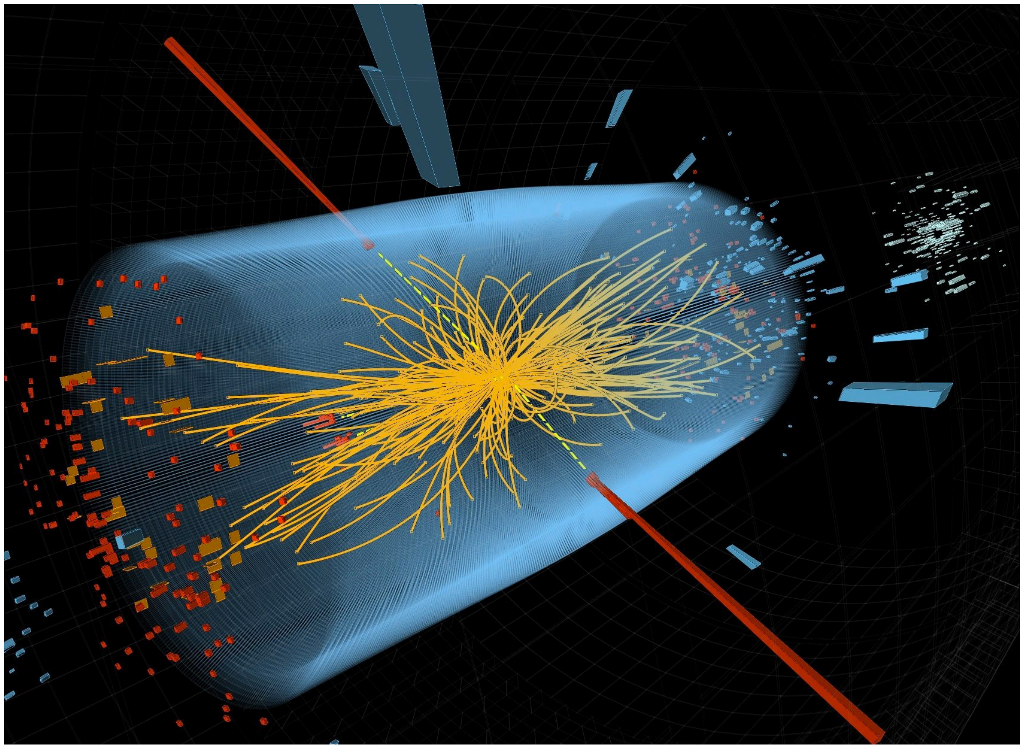


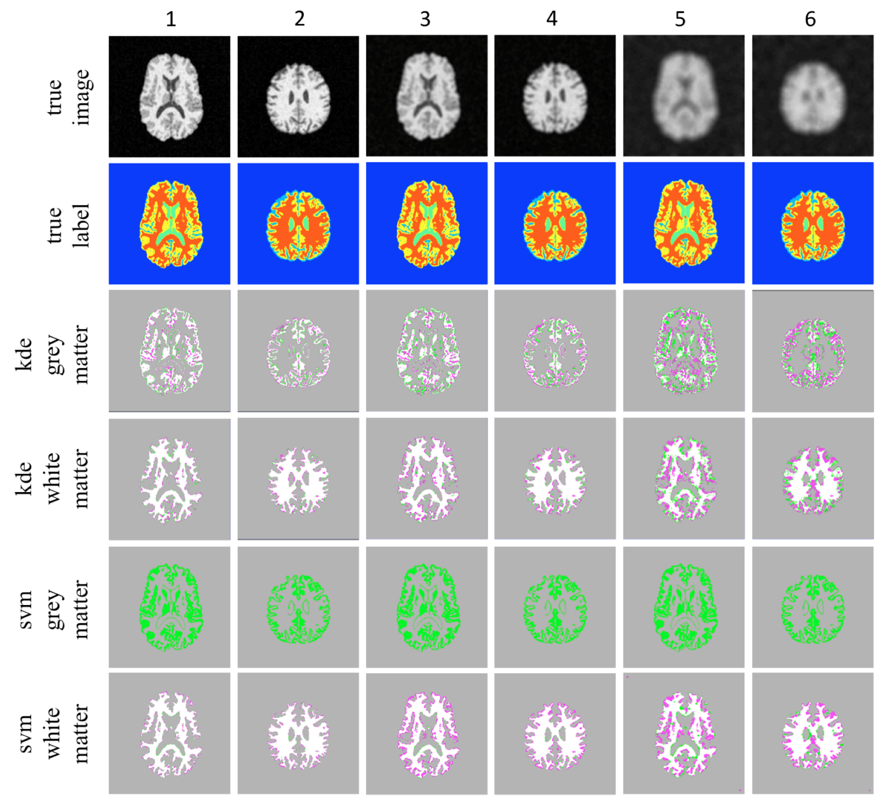
Oscar Bruno, Caltech

# App - Computer graphics/approximation theory

Beatson et al,  SIGGRAPH'01

# App - Supervised learning & kernel machines



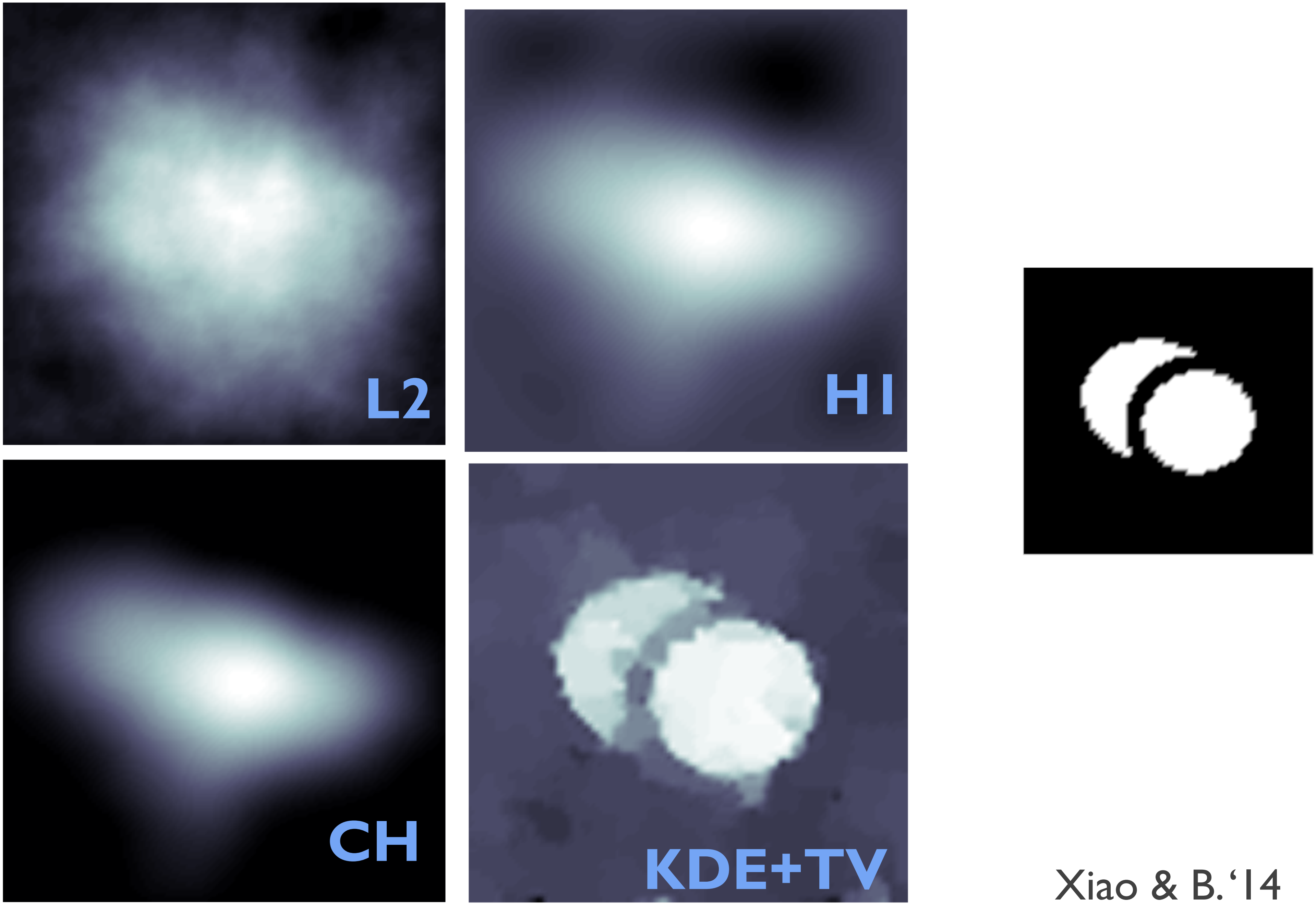



Binary classification

Image segmentation

March, Yu, Xiao, Tharakan , & B, '15

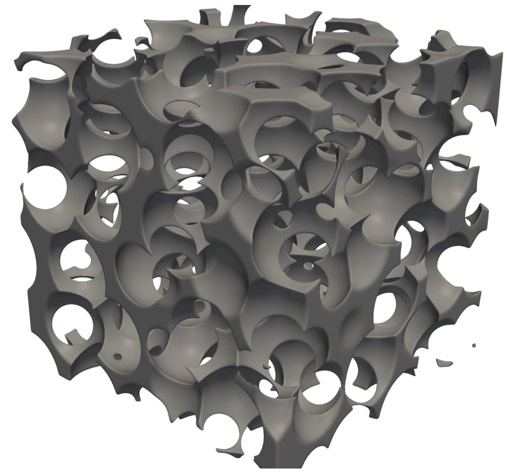
# App - Bayesian priors for inverse problems




Xiao & B. ‘14

# Variable-coefficient problems

## Formulation (Lippmann-Schwinger)



$$u(x) - \Delta u(x) + \eta(x)u(x) = f(x)$$

becomes

$$u(x) + \int_y G(x-y)\eta(y)u(y) = f(x)$$

---

$$-\Delta u(x) - \text{div}\,(\eta(x)\nabla u(x)) = f(x)$$

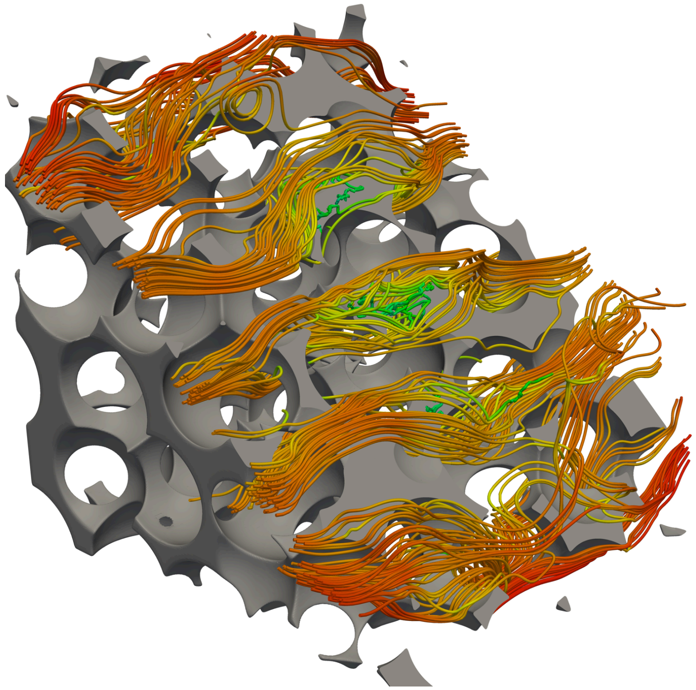
becomes

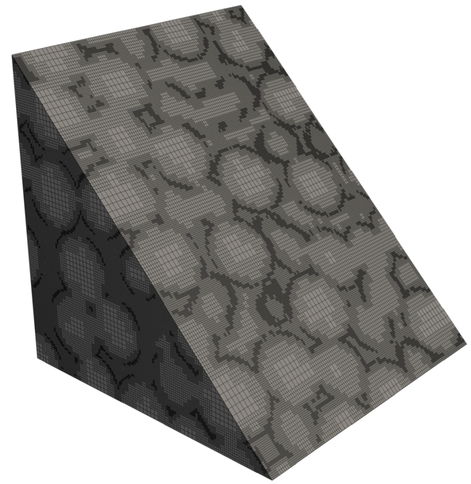
$$u(x) - \int_y \nabla G(x-y)\cdot\eta(y)\nabla u(y) = \int_y G(x-y)f(y)$$

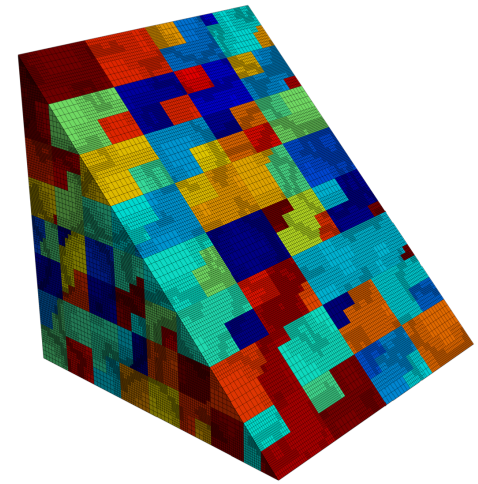
GMRES (PETSc) with volume FMM for the matrix-vector multiplication

# Porous medium flow, 18B dof, jump=1E9

Malhotra, Gholami, B., SC'14







p: Stampede nodes

node: 1.42TFLOP

Xeon: 16 core

Xeon: Phi ~ 22% peak

| $p$ | $N_{\text{dof}}/p$ | $N_{\text{iter}}$ | $T_{solve}$ | TFLOPS | $\eta$ |
|---|---|---|---|---|---|
| 1 | 8.0E+6 | 155 | 477 | 0.36 | 1.00 |
| 6 | 7.8E+6 | 115 | 388 | 2.27 | 1.04 |
| 27 | 8.6E+6 | 101 | 401 | 10.3 | 1.05 |
| 125 | 8.5E+6 | 98 | 419 | 45.3 | 0.99 |
| 508 | 8.9E+6 | 92 | 444 | 173 | 0.94 |
| 2048 | 9.1E+6 | 90 | 474 | 656 | 0.88 |

# ASKIT: N-body in high dimensions

| cores | $T_{\text{skel}}$ | $T_{\text{list}}$ | $T_{\text{let}}$ | $T_{\text{near}}$ | $T_{\text{far}}$ | $T$ |
|---|---|---|---|---|---|---|
| mnist2m $m = 512, s = 128, \kappa = 64, h = 4, \epsilon =$1E-01 | | | | | | |
| 20 | 18 | 11 | 0 | 109 | 592 | 750 |
| 320 | 5 | <1 | 24 | 23 | 16 | 68 |
| 640 | 3 | <1 | 22 | 11 | 8 | 44 |
| mnist2m $m = 512, s = 256, \kappa = 1, h = 4, \epsilon =$1E-01 | | | | | | |
| 20 | 16 | 2 | 0 | 6 | 389 | 413 |
| 320 | 7 | <1 | <1 | 3 | 10 | 20 |
| 640 | 4 | <1 | <1 | 1 | 4 | 10 |
| susy $m = 256, s = 64, \kappa = 256, h = 0.03, \epsilon =$ 1E-04 | | | | | | |
| 20 | 31 | 139 | 0 | 38 | 268 | 476 |
| 320 | 11 | 7 | 36 | 14 | 9 | 78 |
| 640 | 5 | 3 | 19 | 7 | 5 | 39 |
| susy $m = 512, s = 512, \kappa = 64, h = 0.15, \epsilon =$ 1E-01 | | | | | | |
| 20 | 84 | 35 | 0 | 24 | 910 | 1053 |
| 320 | 35 | 1 | 7 | 10 | 30 | 83 |
| 640 | 18 | 1 | 5 | 5 | 15 | 43 |

March, Xiao, B. (arXiv, SISC'15, IPDPS'15)

# Algorithm

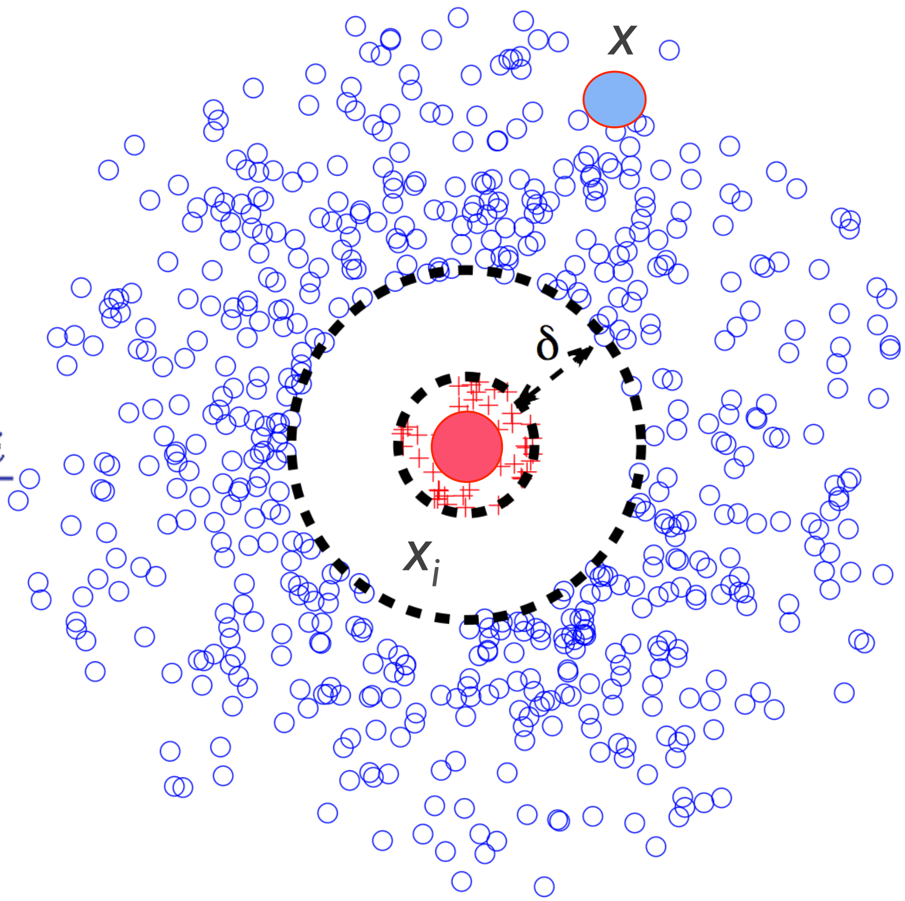
# Barnes-Hut O(N logN)

$x$: Target point $\quad x_i$: source location

$w_i$: weight for sources

$$u(x) = \sum_i G(x, x_i) w_i$$

1. compute $W = \sum_i w_i$
2. compute $x_W = \frac{\sum_i x_i w_i}{W}$
3. $u(x) \approx G(x, x_W) W$



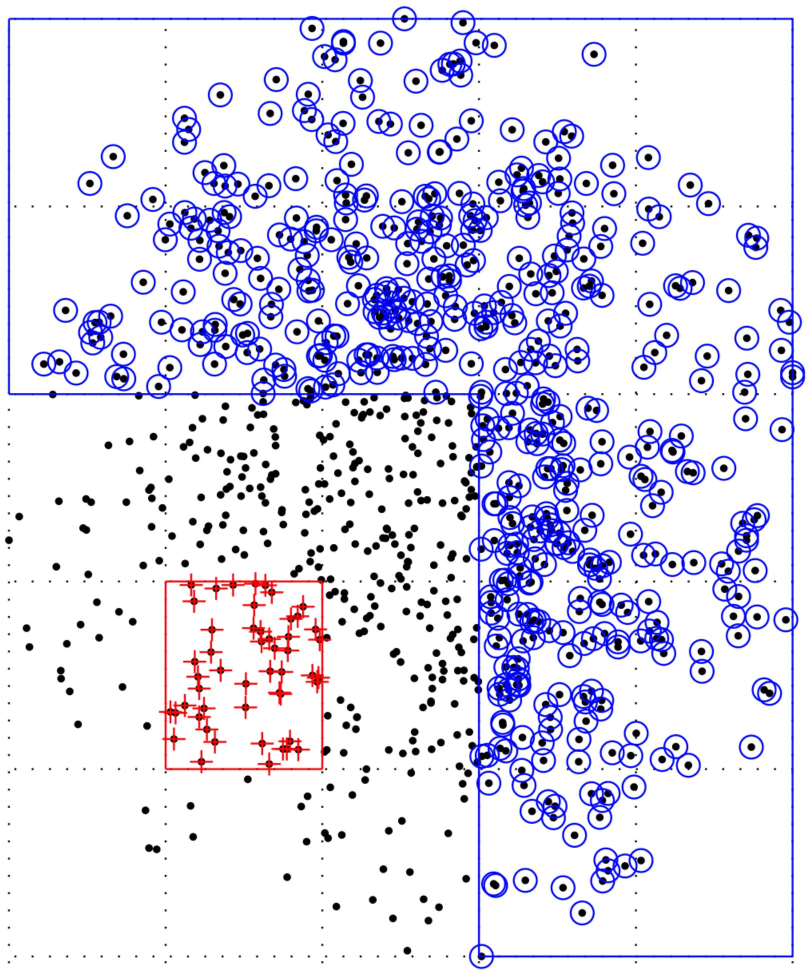
## Questions

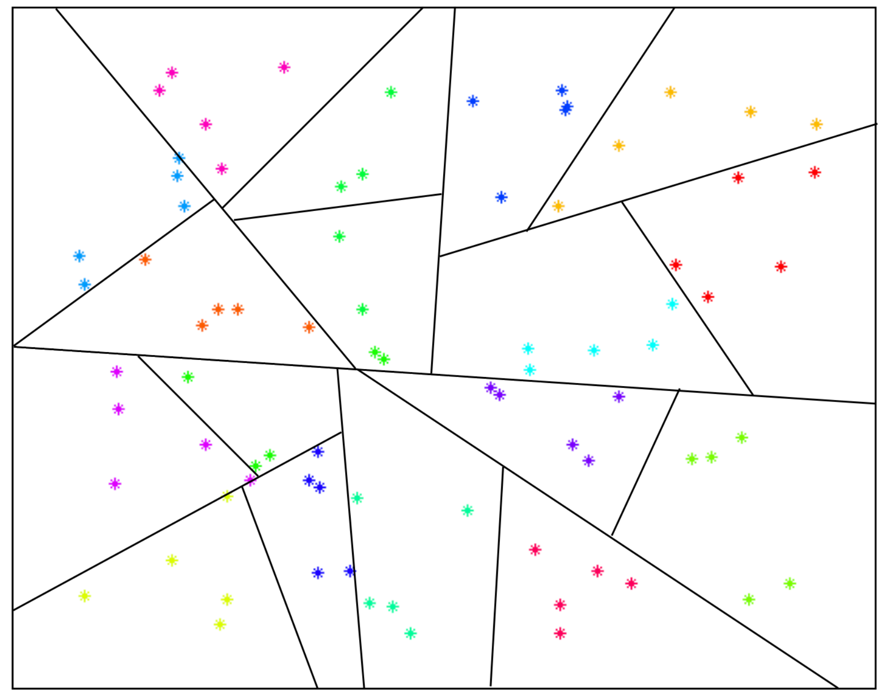
- How do we group points together?
- How do we create more sophisticated far-field representations?
- How to obtain an optimal complexity algorithm?
  - N^2/block → N log N → N
- How do we control accuracy?
- How do we lower the constants?
- How to we parallelize?
- How to we use accelerators?

# Regular decompositions

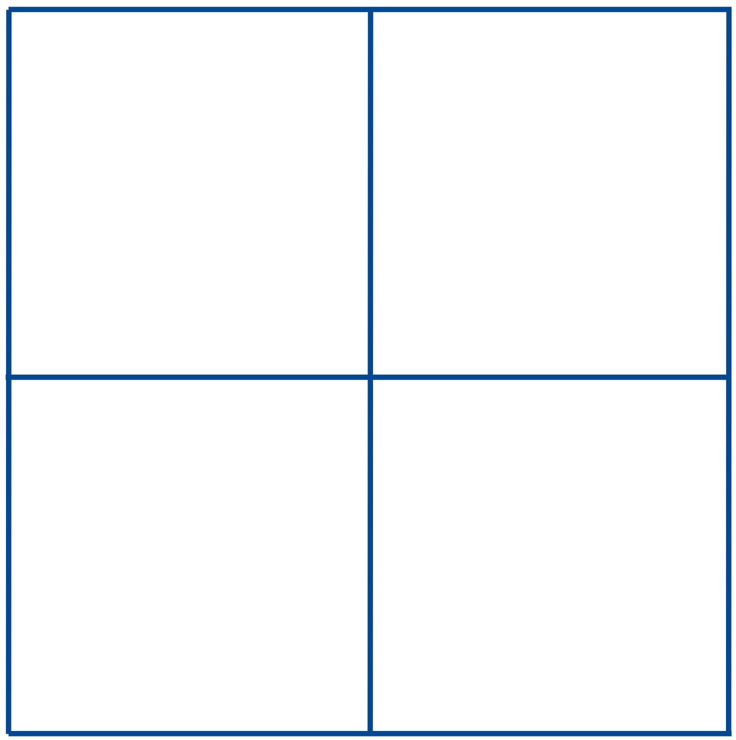
Low dimensions

High dimensions

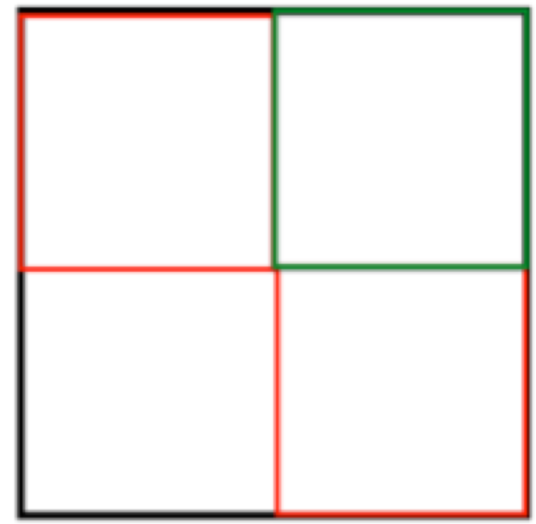


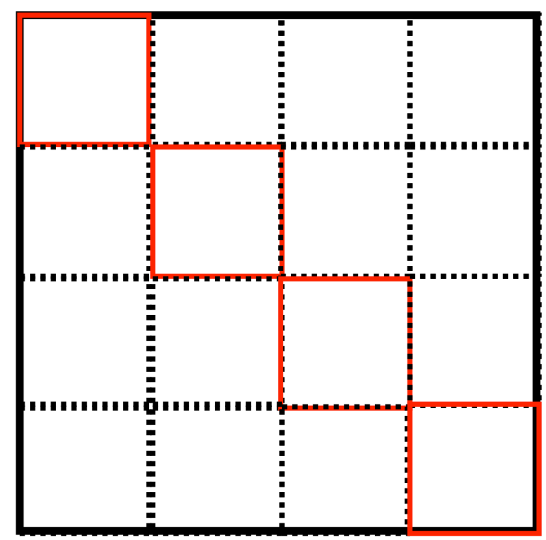


# The algebraic view



$R^d$ (Geometry)

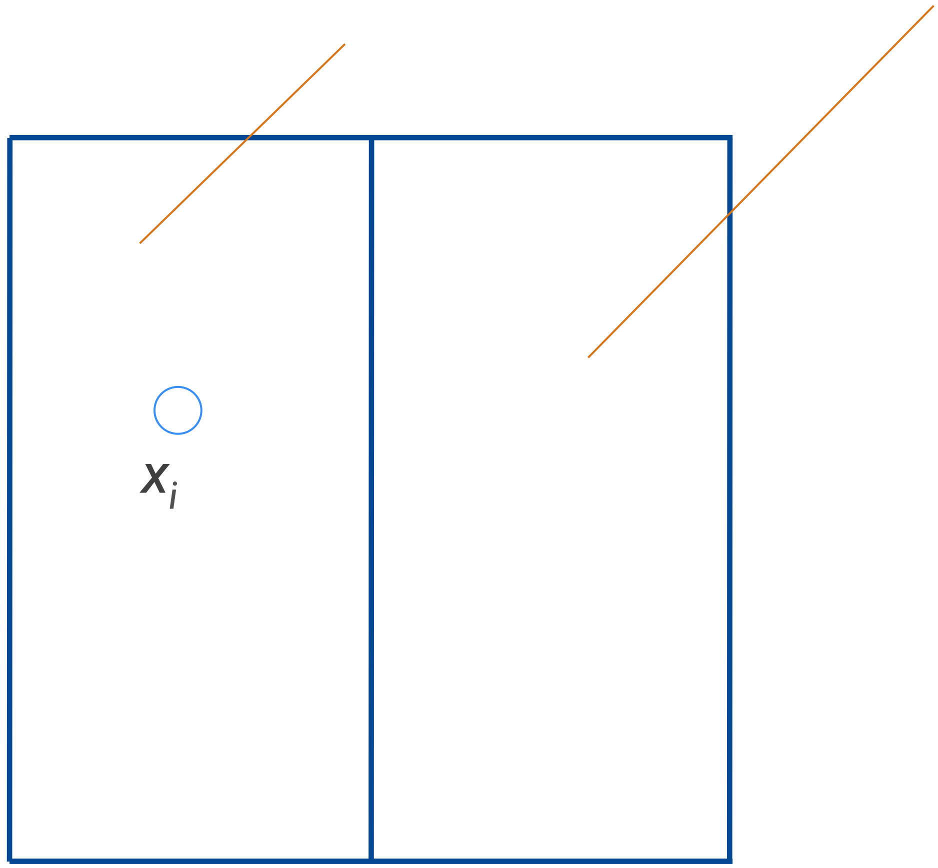




Matrix partitioning

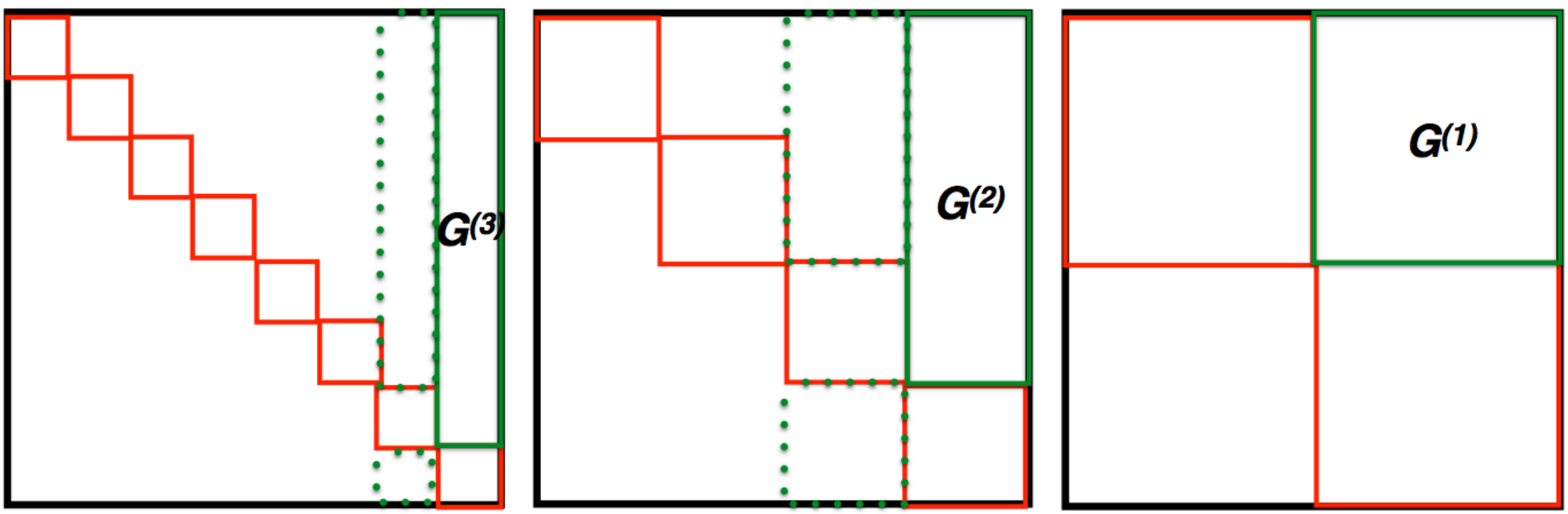
# Geometric view

$$u_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} G(x_i, x_j) w_j = \sum_{j \in \mathrm{near}(i)} G_{ij} w_j + \sum_{j \in \mathrm{far}(i)} G_{ij} w_j$$



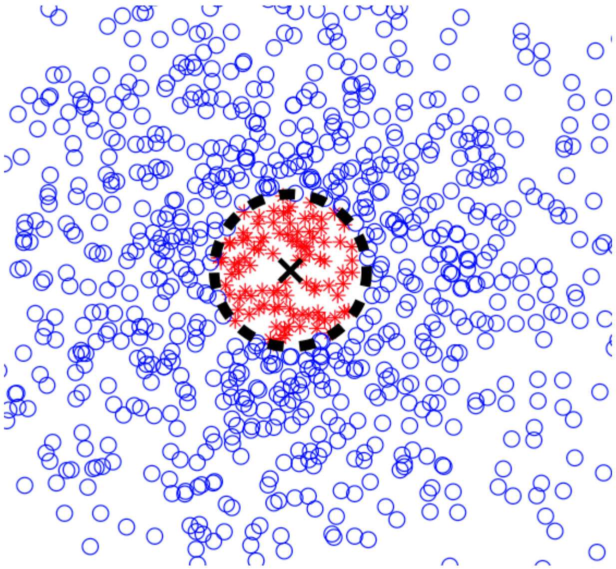
# Algebraic view

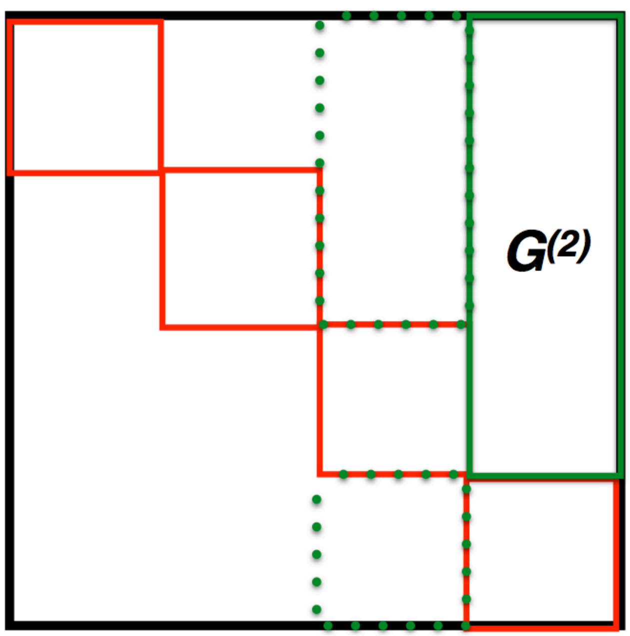
$$u_i = \sum_{j \in \text{diagonal}(i)} G_{ij} w_j + \sum_{j \in \text{off-diagonal}(i)} G_{ij} w_j$$

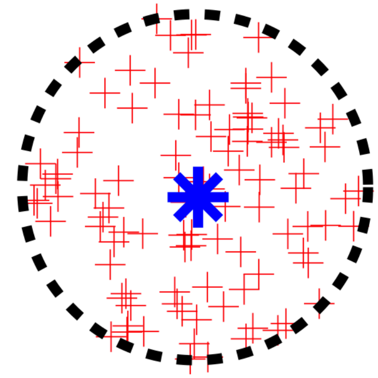


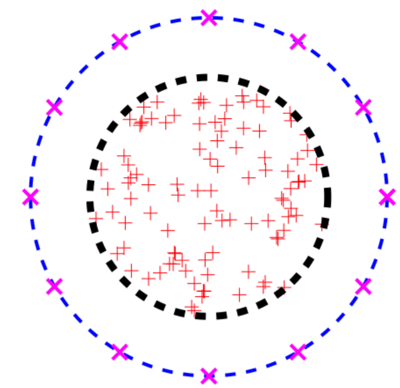

(a) **Level 3.** (b) **Level 2.** (c) **Level 1.**

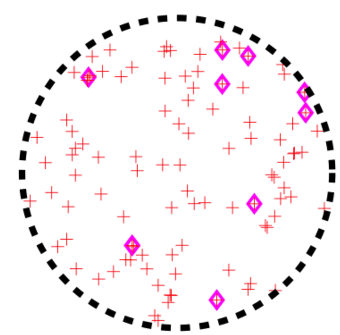
# High order approximations



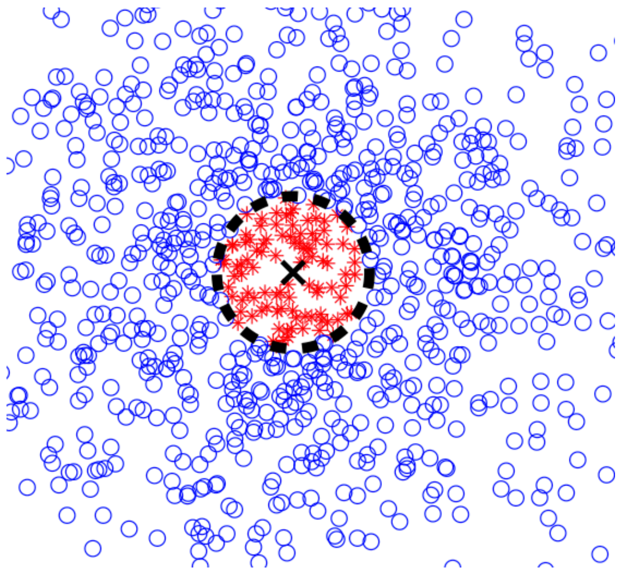


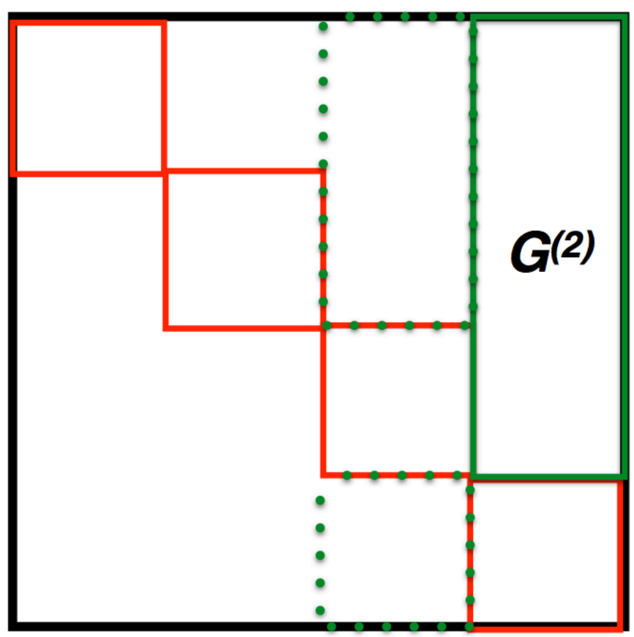







# Low rank approximations






r: numerical rank

SVD/QR: $O(n\ m^2)$

Randomized QR: $O(n\ m\ r)$

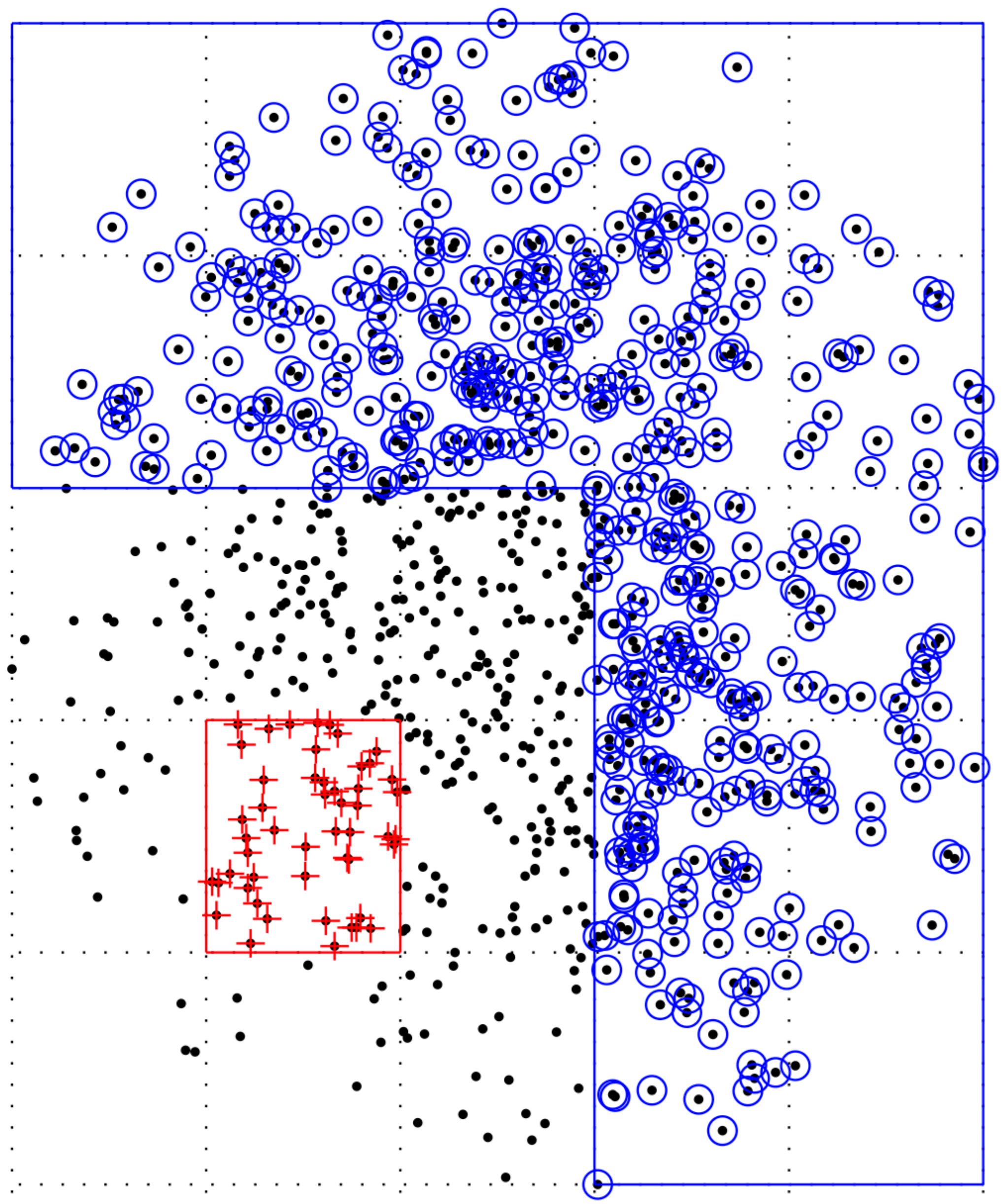
Too expensive.

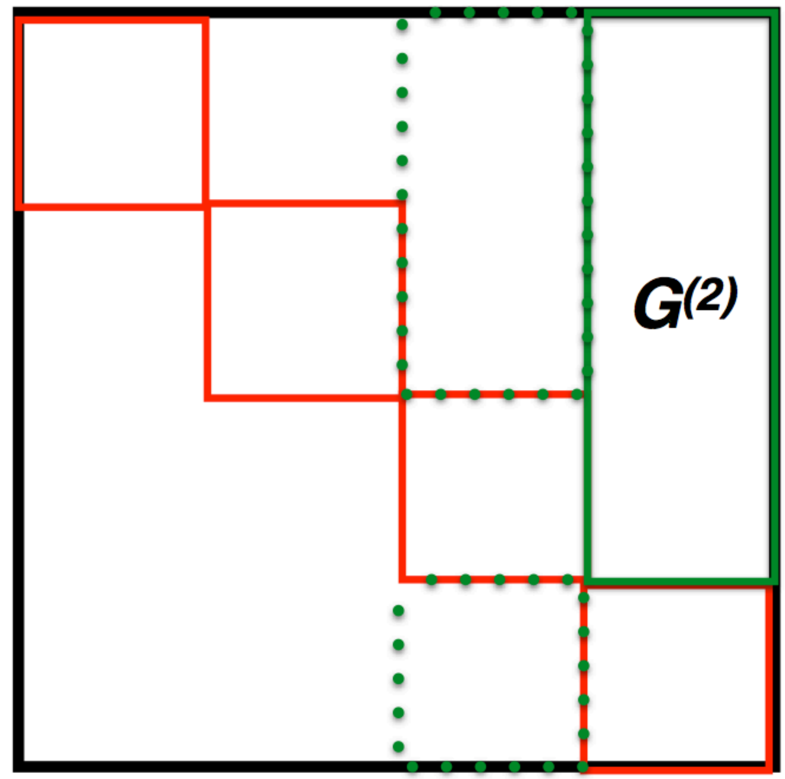
Analytic or semi-analytic

O(r^3) !

But something can be also done for purely algebraic approximations.

# Near field can be expanded (to reduce r)





padas.ices.utexas.edu/software

Open problems:

Solving linear systems

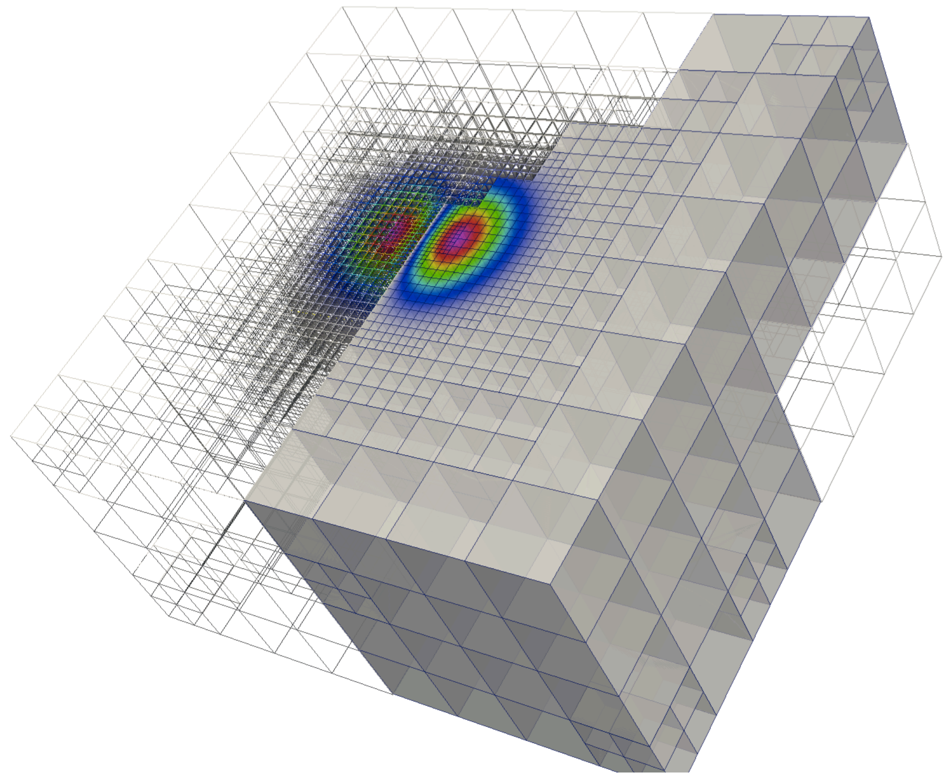
Preconditioners

High-frequency problems

Multiphysics

Optimization/UQ

Time-domain



**Thank you**