Generalized *N*-Body Solvers in the Physical and Information Sciences

Matt Challacombe, Nicolas Bock, Jean Challacombe & Terry Haut
Los Alamos National Laboratory
matt.challacombe@freeon.org

"New Algorithms for Complex Data" March 19th-20th

LA-UR-15-21677
Sponsored by: **DOE LDRD-ER grant 20110230ER** & TenBar Café

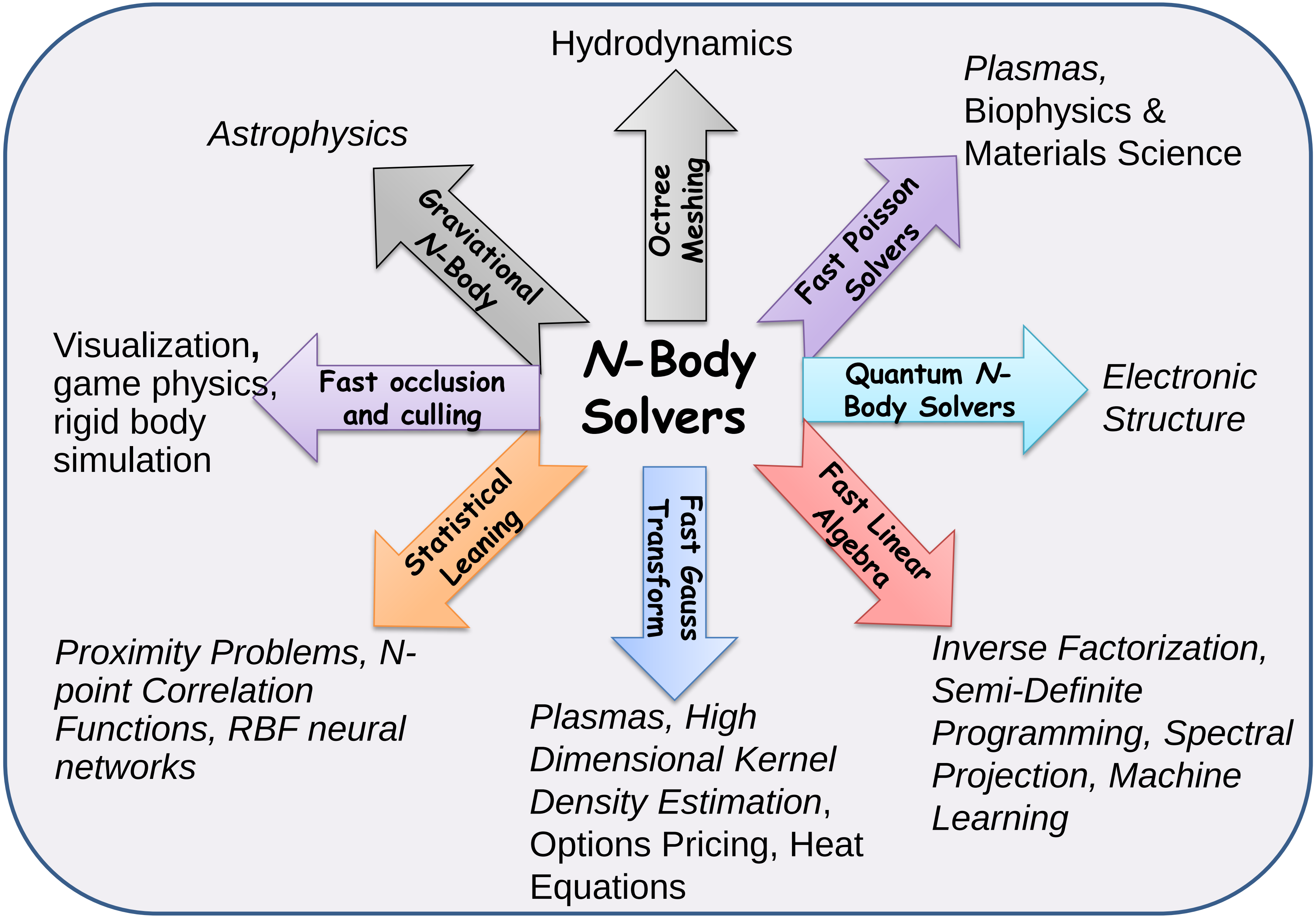
N-Body Solvers in the Physical, Information & Computer Sciences

- *N*-Body solvers combine ***database operations*** (range & metric queries) with ***locality preserving heuristics*** and a wide variety of ***mathematical approximations***. Examples: the astrophysical ***Barnes-Hut tree-code***, the ***Fast Gauss Transform*** and so on.

- The ***generic n-body*** model has been extended to a vast number of fast, pairwise (kernel) summation techniques in the information sciences. Examples: see www.fast-lab.org.

- In ***functional programming***, the *n*-body problem may be developed with the formal properties of ***generacity***, involving map, fold, reduce & *etc*. Examples: the parallel map skeleton, algorithmic skeleton frameworks and so on.

N-Body Solvers You May Know



‘N-Body’ Problems in Statistical Learning

Alexander G. Gray
Depa ment of Computer Science
ie Mellon University
@cs.cmu.edu

Andrew W. Moore
Robotics Inst. and Dept. Comp. Sci.
Carnegie Mellon University
awm@cs.cmu.edu

Massively popularized the generalized *n*-body problem in the information sciences

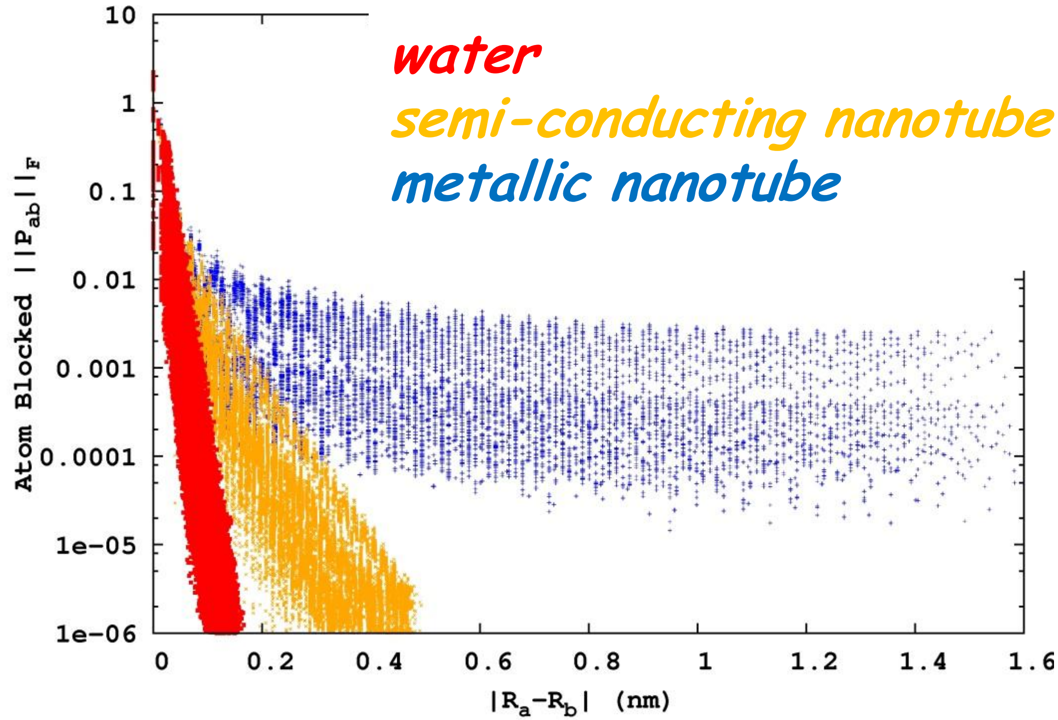
We present very fast algorithms for a large class of statistical problems, which we call all-point-pairs problems, or 'N-body'-like problems. These are problems which abstractly require a comparison of each of the N points in a dataset with each other point and would naively be solved using N^2 computations. Such algorithms cannot be applied to practical problems where N is large. In this paper we focus on several examples of all-point-pairs problems within nonparametric statistical learning: nearest-neighbor classification, kernel density estimation, outlier detection, the two-point correlation, multiple two-point correlation, and the n-point correlation. We give an asymptotic analysis and we show empirically that this algorithmic framework leads to several orders of magnitude in speedup over the naive computation, even for small datasets. We are aware of no exact algorithms for these problems which are faster either empirically or theoretically. The methodology presented in this paper is also applicable in principle to fast, large-scale parametric statistics such as RBF neural networks, mixtures of Gaussians, and Gaussian processes.

Why *N*-Body Algorithms for Complex Data?

- Alignment with enterprise “big data” hardware and software trends: ***n*-body is MapReduce**.
- Potential for extreme simplicity and interoperability in complex scientific applications, *eg.* electronic structure.
- Supported by well developed, simple and commonly available runtime systems such as openmp.
- New mathematical and computational science breakthroughs through focus on ***data locality.***

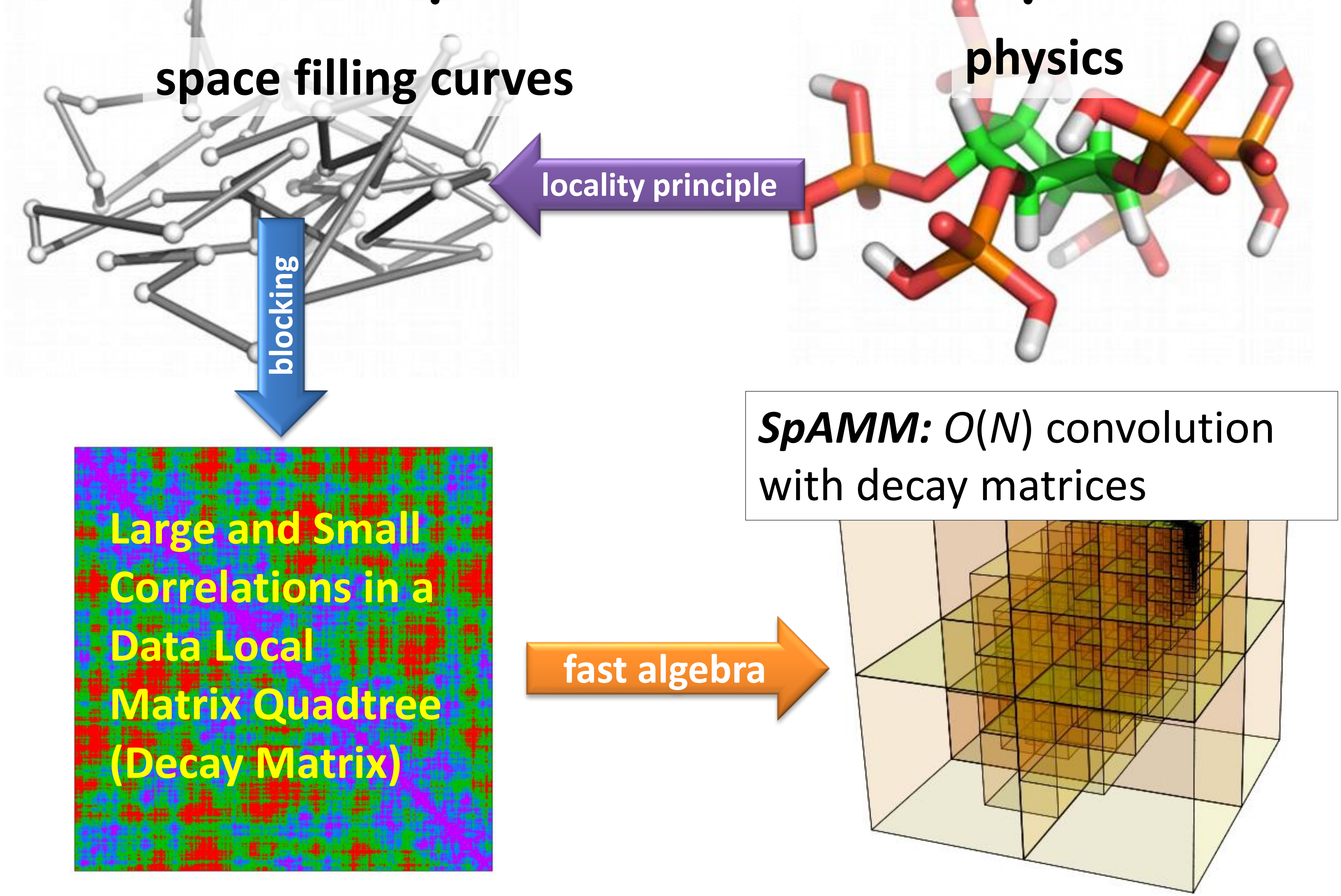
Locality, Decay & Kohn's Nearsighted Principle

Quantum matrices possess decay properties due to the finite range of quantum interactions for non-metallic systems.



We are interested in systems with metric (basis) and gap ill-conditioning, leading to slow rates of decay. These problems occur with accurate methods and for problems involving strong correlation.

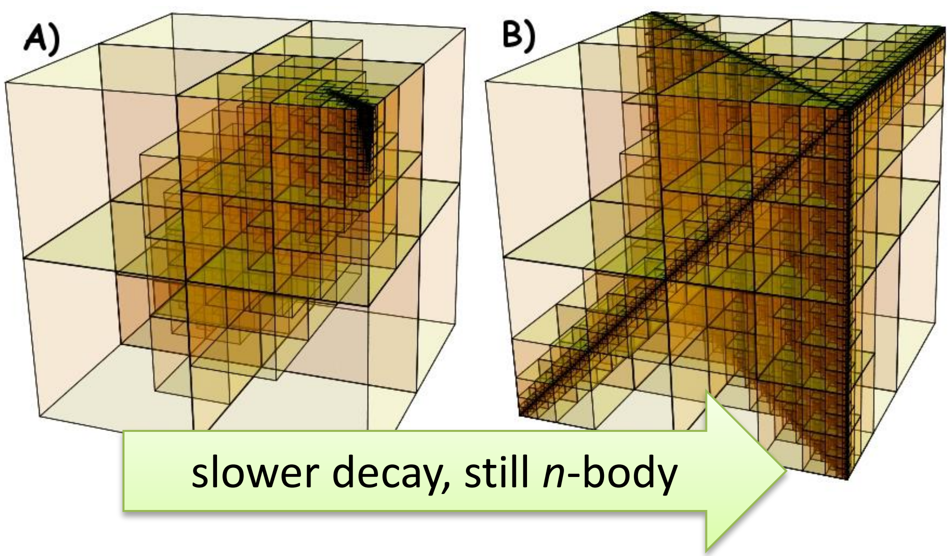
Matrix Multiplication as N-Body Solver



Matrix Multiplication as *N*-Body Solver

SpAMM is a fast kernel for multiplication of matrices with decay & structured locality. ***SpAMM is recursive occlusion and culling based on the Cauchy-Schwarz inequality, applied to data local matrix quadtrees:***

$$A^k \otimes B^k = \begin{cases} 0 & \text{if } \left\|A^k\right\| \left\|B^k\right\| < \tau, \\ A^k \cdot B^k & \text{elseif } k = k_{\max}, \\ \begin{pmatrix} A_{11}^{k+1} \otimes B_{11}^{k+1} + A_{12}^{k+1} \otimes B_{21}^{k+1} & A_{11}^{k+1} \otimes B_{12}^{k+1} + A_{12}^{k+1} \otimes B_{22}^{k+1} \\ A_{21}^{k+1} \otimes B_{11}^{k+1} + A_{22}^{k+1} \otimes B_{21}^{k+1} & A_{21}^{k+1} \otimes B_{21}^{k+1} + A_{22}^{k+1} \otimes B_{22}^{k+1} \end{pmatrix} & \text{else.} \end{cases}$$



A) Exponential Decay
B) Algebraic Decay

Groups+Perturbations, a SpAMM Algebra

SpAMM yields a biased Lie Algebra:

$$[\boldsymbol{a}\,,\boldsymbol{b}]_\tau = \boldsymbol{a} \otimes_\tau \boldsymbol{b} - \boldsymbol{b} \otimes_\tau \boldsymbol{a} = \tau\, \widehat{\boldsymbol{\delta}}$$

with perturbed vector field $\tau\widehat{\boldsymbol{\delta}}$.

Unlike round-off, the SpAMM bias is deterministic, resulting from the branch pattern set by Cauchy-Schwarz occlusion and culling.

Interested in the way important iterations are affected by large τ approximations (preconditiong!).

Newton Schulz for $s^{-1/2}$ and Sign(s)

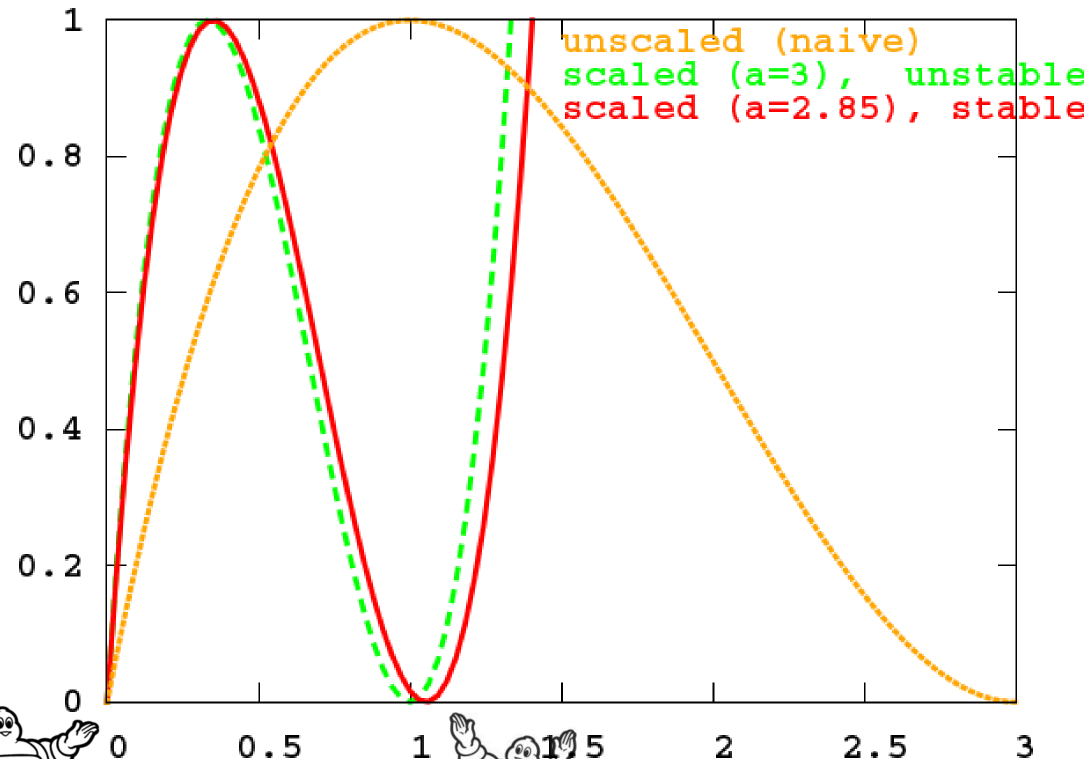
- Most important matrix functions for Electronic Structure.
- Can be very ill-conditioned for high quality theories, with long-range correlation and large basis sets
- $\mathrm{sign}(\boldsymbol{s})$ and $\boldsymbol{s}^{-1/2}$ are related by Higham's identity (Higham '97):

$$\mathrm{sign}\left(\begin{bmatrix} 0 & \boldsymbol{s} \\ \boldsymbol{I} & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & \boldsymbol{s}^{1/2} \\ \boldsymbol{s}^{-1/2} & 0 \end{bmatrix}$$

- Many variations in literature. Error analysis focused almost entirely on the NS map.

Scaled Newton Schulz

- The naïve NS map is: $\mathrm{m}[x] \coloneqq \frac{1}{2}(3-x)$, corresponding to the logistic mapping $x \leftarrow \frac{1}{2}(3-x) \cdot x \cdot \frac{1}{2}(3-x)$. Much recent work on the scaled NS. See for example Pan & Schreiber '91, Higham '97, Janzik et al '07, ***Chen & Chow '14***.

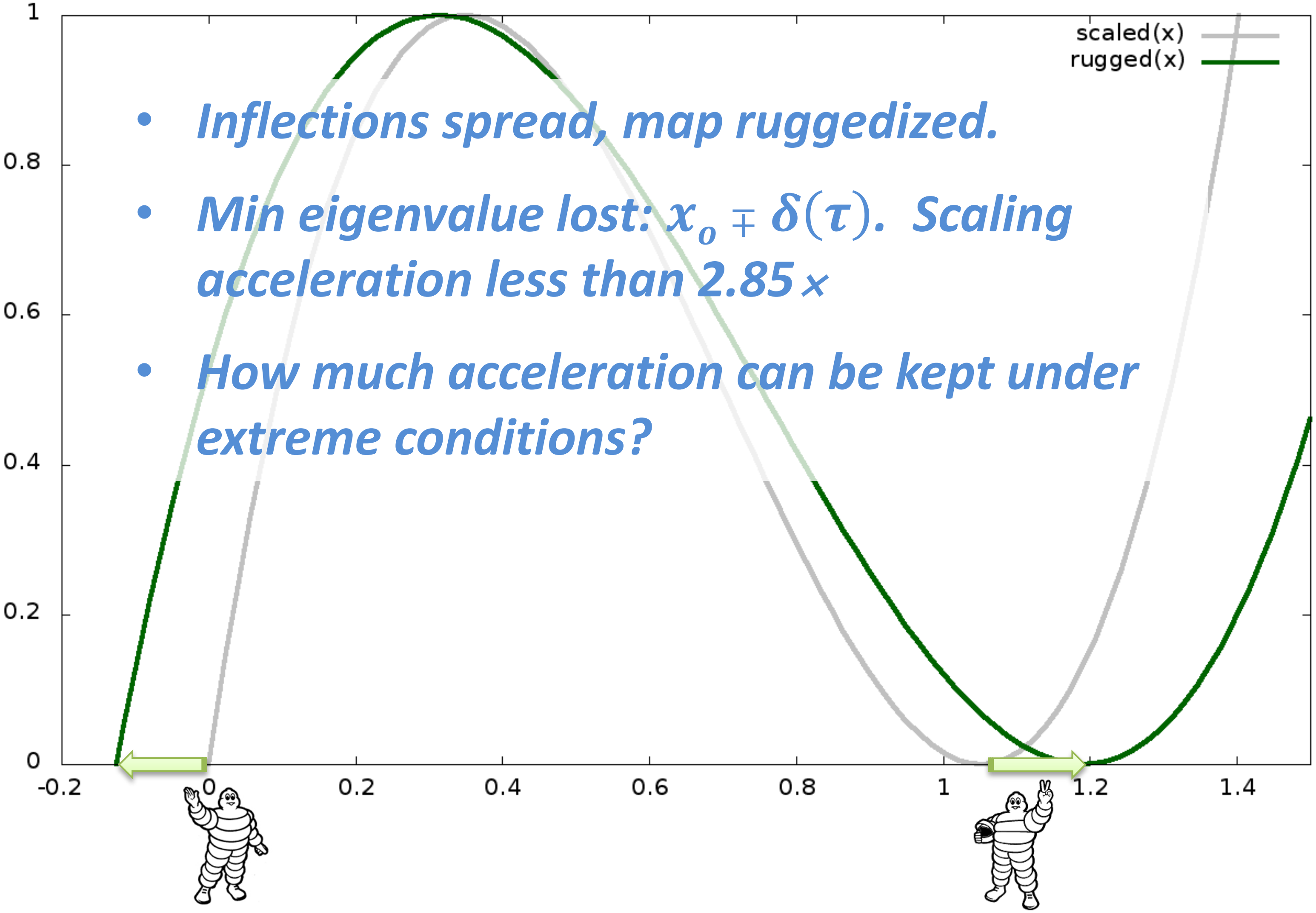


Scaling accelerates convergence by increasing the map gradient @ min EV: $\mathrm{g}(x_0) = \boldsymbol{\alpha} \times \left(\frac{9}{4}\right)$.

Using $\alpha \sim 2.85$ instead of $\alpha = 3$ stabilizes map for double precision

A Ruggedized, Scaled NS Map

Ill-conditioning and SpAMM can bounce EVs out of bounds by $\mp\,\delta(\tau)$. ⇨ stabilize by spreading 0/1 inflections by $\sim \mp\,\delta$.



- *Inflections spread, map ruggedized.*
- *Min eigenvalue lost: $x_o \mp \delta(\tau)$. Scaling acceleration less than $2.85\times$*
- *How much acceleration can be kept under extreme conditions?*

Bias Stability Under SpAMM Iteration

Consider iteration under (perhaps complicated) map and fold with the SpAMM algebra:

$$\widetilde{\boldsymbol{x}}_k \leftarrow f(\widetilde{\boldsymbol{x}}_{k-1})$$

At each iteration *k* there is a bias, $\delta \boldsymbol{x}_k$, of order τ:

$$\widetilde{\boldsymbol{x}}_k = \boldsymbol{x}_k + \tau \, \widehat{\delta \boldsymbol{x}}_k$$

The fate of this bias is determined by the Gateaux differential,

$$\boldsymbol{f}_{\widehat{\delta x}} = \lim_{\tau \to 0} \left\{ \frac{f(\boldsymbol{x} + \tau \, \widehat{\delta \boldsymbol{x}}) - f(\boldsymbol{x})}{\tau} \right\}, \quad \text{via} \quad \delta \boldsymbol{x}_k \leftarrow \tau \, \boldsymbol{f}_{\widehat{\delta x}_{k-1}}.$$

Then, the iteration is stable if $\left\| \boldsymbol{f}_{\widehat{\delta x}} \right\| < 1$.

Scaled Newton Schulz Iterations (Three)

Newton Schulz iteration ($\mathbf{ns}$) with map $\mathtt{m}[\cdot]$ and $\otimes_\tau$ algebra:

$\{\boldsymbol{x}_\tau, \boldsymbol{z}_\tau\} \leftarrow \mathbf{ns}[\boldsymbol{s}, \tau]$, with $\boldsymbol{x}_\tau \to \boldsymbol{I}$ and $\boldsymbol{z}_\tau \to \boldsymbol{s}^{-1/2}$ as $\tau \to 0$

There are (at least) three versions, ***naïve, stable*** and ***dual channel***:

$$\mathbf{ns}^{\mathrm{naiv}}[\boldsymbol{s}, \tau] \coloneqq \begin{array}{c} \boldsymbol{x}_0 = \boldsymbol{s}, \boldsymbol{z}_0 = \boldsymbol{I} \\ \mathtt{while}\ (\,|tr\boldsymbol{x}_k - n|/n > \tau\,) \\ \mathtt{return}\ \{\boldsymbol{x}_\tau \leftarrow \boldsymbol{x}_k, \boldsymbol{z}_\tau \leftarrow \boldsymbol{z}_k\} \end{array} \begin{bmatrix} \boldsymbol{z}_k \leftarrow \boldsymbol{z}_{k-1} \otimes_\tau \boldsymbol{x}_{k-1} \\ \boldsymbol{x}_k \leftarrow \boldsymbol{z}_k \otimes_\tau \boldsymbol{s} \otimes_\tau \boldsymbol{z}_k \end{bmatrix}$$

$$\mathbf{ns}^{\mathrm{stab}}[\boldsymbol{s}, \tau] \coloneqq \begin{array}{c} \boldsymbol{x}_0 = \boldsymbol{s}, \boldsymbol{z}_0 = \boldsymbol{I} \\ \mathtt{while}\ (\,|tr\boldsymbol{x}_k - n|/n > \tau\,) \\ \mathtt{return}\ \{\boldsymbol{x}_\tau \leftarrow \boldsymbol{x}_k, \boldsymbol{z}_\tau \leftarrow \boldsymbol{z}_k\} \end{array} \begin{bmatrix} \boldsymbol{z}_k \leftarrow \boldsymbol{z}_{k-1} \otimes_\tau \boldsymbol{x}_{k-1} \\ \boldsymbol{x}_k \leftarrow \boldsymbol{z}^T{}_k \otimes_\tau \boldsymbol{s} \otimes_\tau \boldsymbol{z}_k \end{bmatrix}$$

$$\mathbf{ns}^{\mathrm{dual}}[\boldsymbol{s}, \tau] \coloneqq \begin{array}{c} \boldsymbol{x}_0 = \boldsymbol{s}, \boldsymbol{y}_0 = \boldsymbol{s}, \boldsymbol{z}_0 = \boldsymbol{I} \\ \mathtt{while}\ (\,|tr\boldsymbol{x}_k - n|/n > \tau\,) \\ \mathtt{return}\ \{\boldsymbol{x}_\tau \leftarrow \boldsymbol{x}_k, \boldsymbol{z}_\tau \leftarrow \boldsymbol{z}_k\} \end{array} \begin{bmatrix} \boldsymbol{z}_k \leftarrow \boldsymbol{z}_{k-1} \otimes_\tau \mathtt{m}[\boldsymbol{x}_{k-1}] \\ \boldsymbol{y}_k \leftarrow \mathtt{m}[\boldsymbol{x}_{k-1}] \otimes_\tau \boldsymbol{y}_k \\ \boldsymbol{x}_k \leftarrow \boldsymbol{y}_k \otimes_\tau \boldsymbol{z}_k \end{bmatrix}$$

Gateaux Differentials and Stability

Recall: $\delta\boldsymbol{x}_k \leftarrow \tau\, \boldsymbol{f}_{\widehat{\delta\boldsymbol{x}}_{k-1}}$. Stability is determined by terms in braces, which tend to identity, $\{\cdot\} \rightarrow \boldsymbol{I}$.

$$\boldsymbol{f}_{\widehat{\delta\boldsymbol{x}}_{k-1}}{}^{\text{naiv}} = \boxed{\tilde{\boldsymbol{z}}_{k-1} \cdot \mathrm{m}'\left[\boldsymbol{x}_{k-1}\right] \cdot \widehat{\delta\boldsymbol{x}}_{k-1} \cdot \boldsymbol{s} \cdot \tilde{\boldsymbol{z}}_k} + \left\{\tilde{\boldsymbol{z}}_k{}^T \cdot \boldsymbol{s} \cdot \tilde{\boldsymbol{z}}_{k-1}\right\} \cdot \mathrm{m}'\left[\boldsymbol{x}_{k-1}\right] \cdot \widehat{\delta\boldsymbol{x}}_{k-1}$$

This term is pathological.

$$\boldsymbol{f}_{\widehat{\delta\boldsymbol{x}}_{k-1}}{}^{\text{stab}} = \widehat{\delta\boldsymbol{x}}_{k-1}{}^T \cdot \mathrm{m}'\left[\boldsymbol{x}_{k-1}\right] \cdot \boxed{\left\{\tilde{\boldsymbol{z}}_{k-1}{}^T \cdot \boldsymbol{s} \cdot \tilde{\boldsymbol{z}}_k\right\}} + \boxed{\left\{\tilde{\boldsymbol{z}}_k{}^T \cdot \boldsymbol{s} \cdot \tilde{\boldsymbol{z}}_{k-1}\right\}} \cdot \mathrm{m}'\left[\boldsymbol{x}_{k-1}\right] \cdot \widehat{\delta\boldsymbol{x}}_{k-1}$$

These terms tend to $\boldsymbol{I}$ and remain near $\boldsymbol{s}$.

$$\boldsymbol{f}_{\widehat{\delta\boldsymbol{x}}_{k-1}}{}^{\text{dual}} = \mathrm{m}'\left[\boldsymbol{x}_{k-1}\right] \cdot \widehat{\delta\boldsymbol{x}}_{k-1} \cdot \boxed{\left\{\tilde{\boldsymbol{y}}_{k-1} \cdot \tilde{\boldsymbol{z}}_k\right\}} + \boxed{\left\{\tilde{\boldsymbol{y}}_k \cdot \tilde{\boldsymbol{z}}_{k-1}\right\}} \cdot \mathrm{m}'\left[\boldsymbol{x}_{k-1}\right] \cdot \widehat{\delta\boldsymbol{x}}_{k-1}$$

These terms should tend to $\boldsymbol{I}$, *but have problems b/c they do not remain near* $\boldsymbol{s}$.

SpAMM Stabilized Scaled NS

- In addition to theoretical development, many experiments to find the most stability under the most extreme permisive approximation.

- Found use of the left transpose SpAMM, ${}^{T}\otimes_{\tau}$, to be most stable, in combination with a tighter first product.

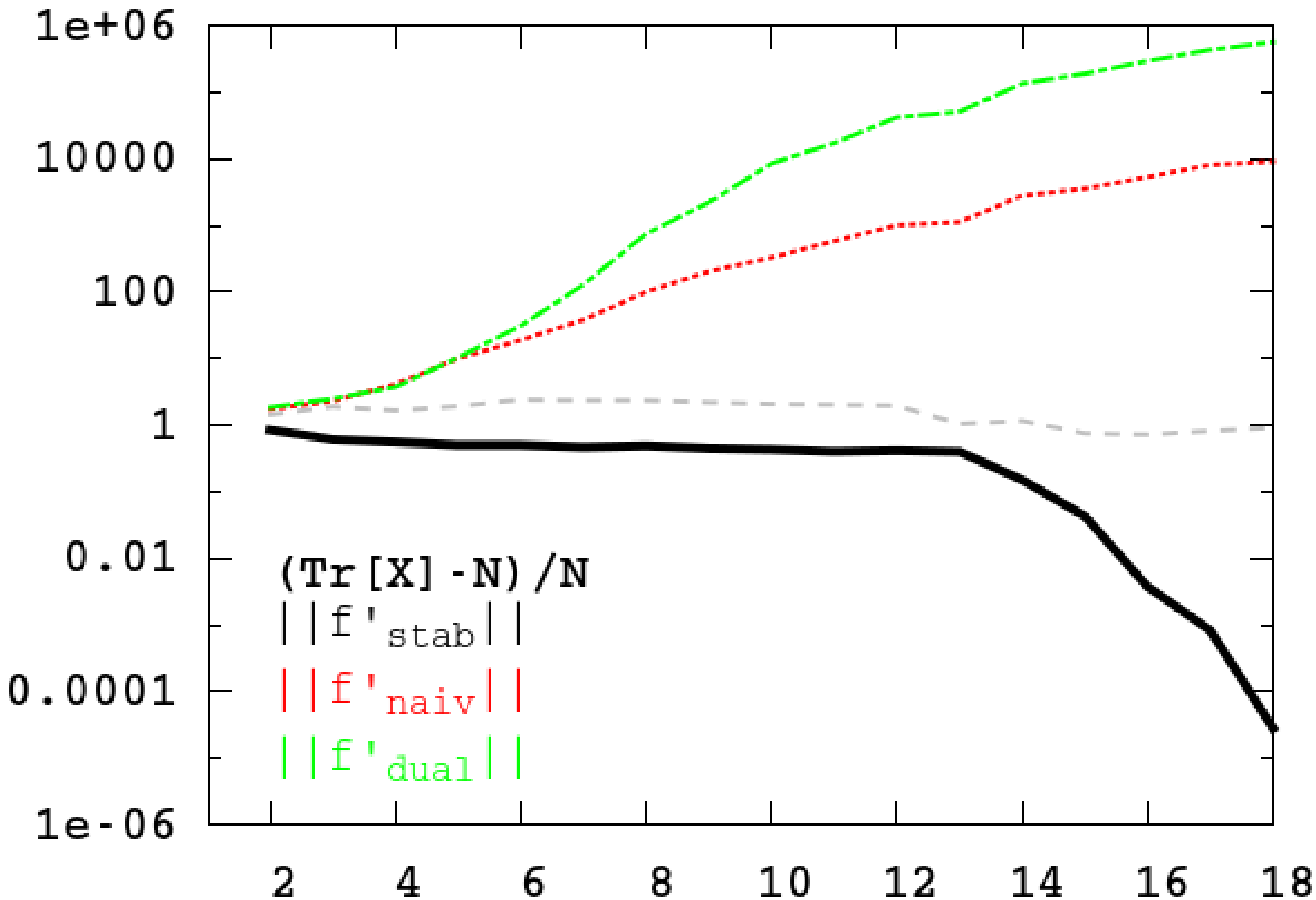
$$\boldsymbol{x}_{k+1} \leftarrow \langle \boldsymbol{z}^{T}{}_{k} | \, \boldsymbol{s} \, | \boldsymbol{z}_{k} \rangle$$

then right: $\boldsymbol{x} \leftarrow \boldsymbol{a} \otimes_{\tau} \boldsymbol{z}$

left (T) first: $\boldsymbol{a} \leftarrow \boldsymbol{z} \, {}^{T}\otimes_{\tau\times 10^{-2}} \, \boldsymbol{s}$

- These approximations yield stability for extreme loose SpAMM approximation in NS, with $\tau = 10^{-3}$, 8×8 blocking, even for extreme ill-conditioned problems to $\boldsymbol{\kappa}(\boldsymbol{s}) = 10^{11}$.

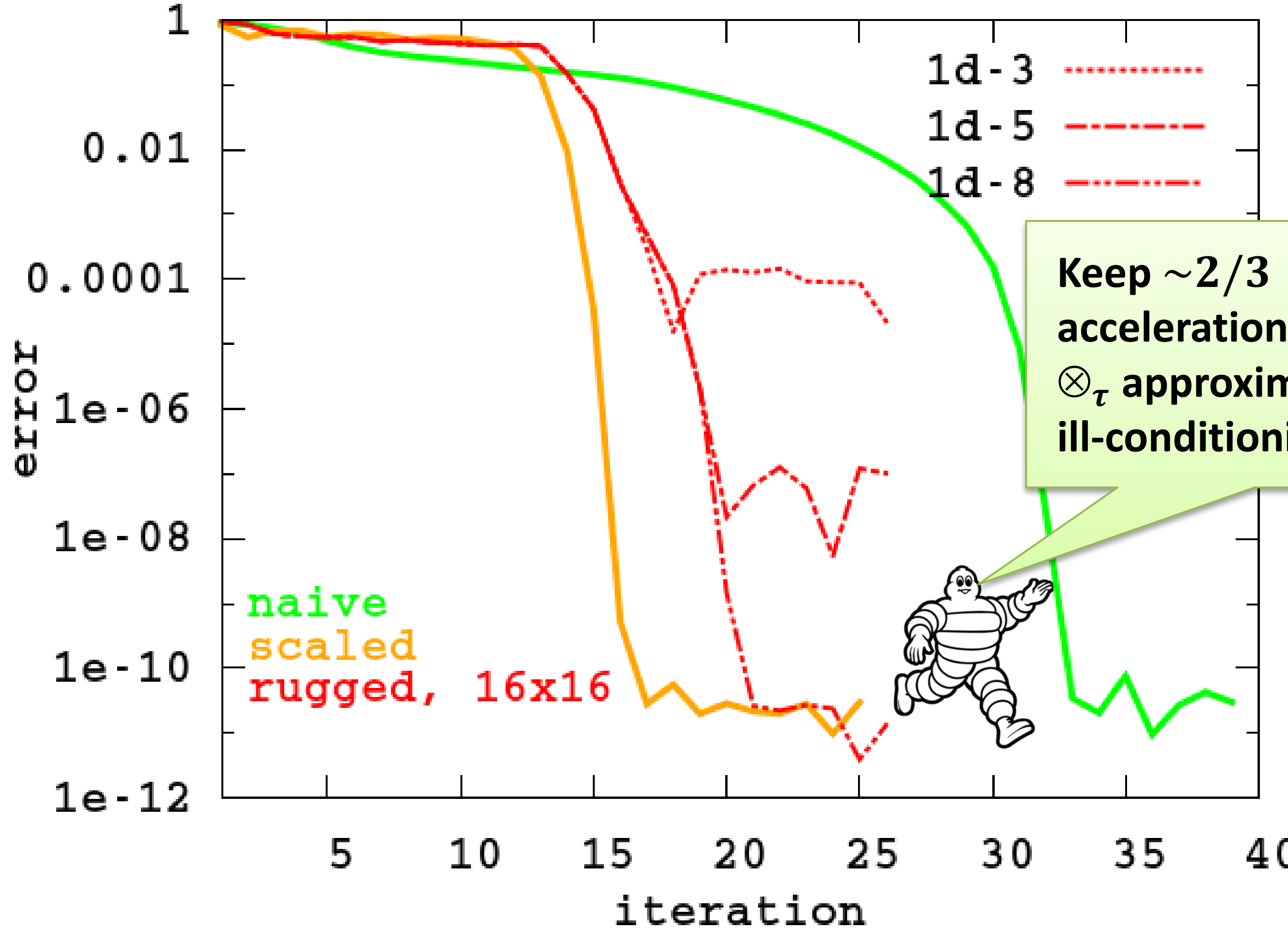
Ill-Conditioning: $\kappa(s) = 10^{11}$, (3,3)x8 nanotube



Convergence of extreme loose SpAMM ($\tau = 5 \times 10^{-3}$, 8x8 blocking) + extreme ill-conditioned problem (thick, black line). Also shown are Gateaux differentials for stabilized, naïve and dual iterations.

SpAMM Stabilized Scaled NS

- Extreme Ill-Conditioning: $\boldsymbol{\kappa}(s) = 10^{11}$, (3,3) x 8 nanotube
- Stabilized, left transpose SpAMM, stabilized map, & scaling switched by error heuristics (don't compute x_0).



Keep $\sim 2/3$ of scaling acceleration with extreme $\otimes_\tau$ approximation & ill-conditioning!

Recursive Preconditioning: The SpAMM Sandwich

Nested Newton Shulz functionals with increasing SpAMM resolution, $\tau_m < \tau_{m-1} < \cdots < \tau_0$. All the work goes into the first step; get full precision for close to free.

$$\{\boldsymbol{x}_{\tau_0}, \boldsymbol{z}_{\tau_0}\} \leftarrow \mathbf{ns}[\boldsymbol{s}, \tau_0]$$

$$\{\boldsymbol{x}_{\tau_1}, \boldsymbol{z}_{\tau_1}\} \leftarrow \mathbf{ns}\left[\{\boldsymbol{z}^T{}_{\boldsymbol{\tau}_0} \otimes_{\tau_1} \boldsymbol{s} \otimes_{\tau_1} \boldsymbol{z}_{\boldsymbol{\tau}_0}\}, \tau_1\right]$$

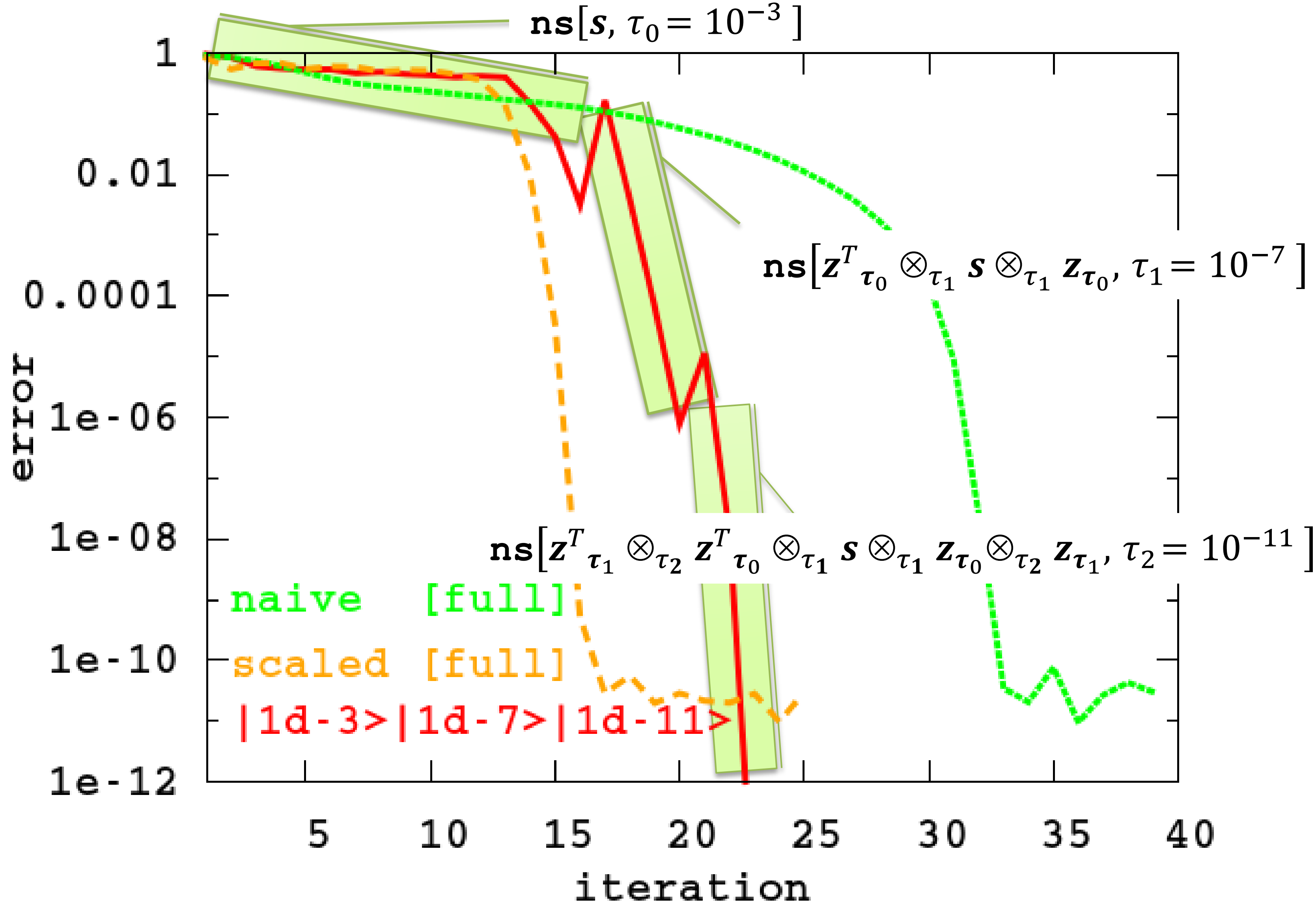
$$\vdots$$

$$\{\boldsymbol{x}_{\tau_m}, \boldsymbol{z}_{\tau_m}\} \leftarrow \mathbf{ns}\left[\{\boldsymbol{z}^T{}_{\boldsymbol{\tau}_{m-1}} \otimes_{\boldsymbol{\tau}_m} \cdots \boldsymbol{s} \cdots \otimes_{\boldsymbol{\tau}_m} \boldsymbol{z}_{\boldsymbol{\tau}_{m-1}}\}, \tau_m\right]$$

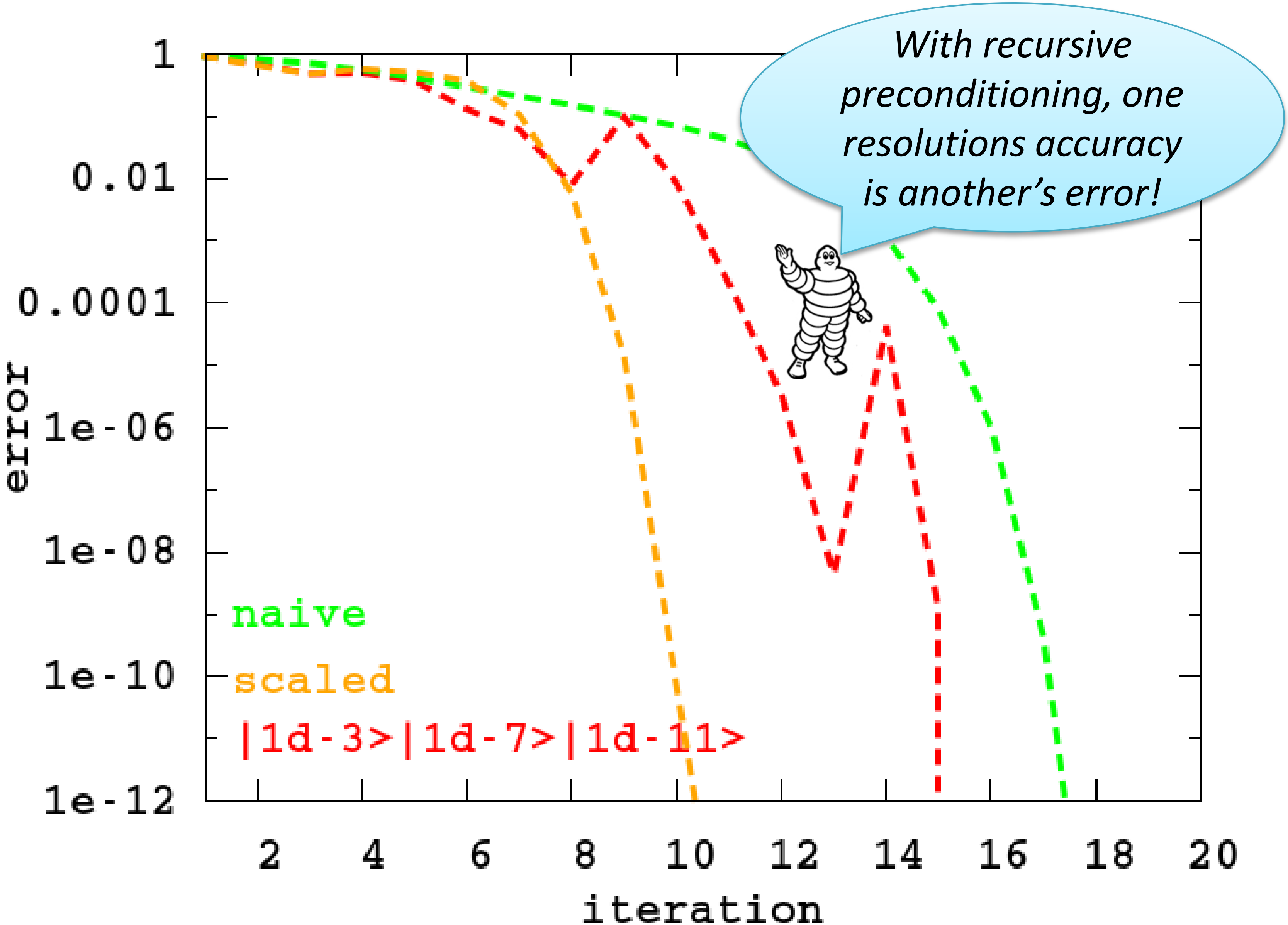
Then, the inverse square root is

- $$\boldsymbol{s}^{-1/2} = \boldsymbol{z}_{\tau_m} \otimes_{\boldsymbol{\tau}_m} \boldsymbol{z}_{\tau_{m-1}} \cdots \otimes_{\boldsymbol{\tau}_1} \boldsymbol{z}_{\tau_0}$$

Ill-Conditioning: $\kappa(s) = 10^{11}$, (3,3)x8 nanotube



Ill-Conditioning: $\kappa(s) \sim 10^5$, $[H_2O]_{70}$, TZV2P

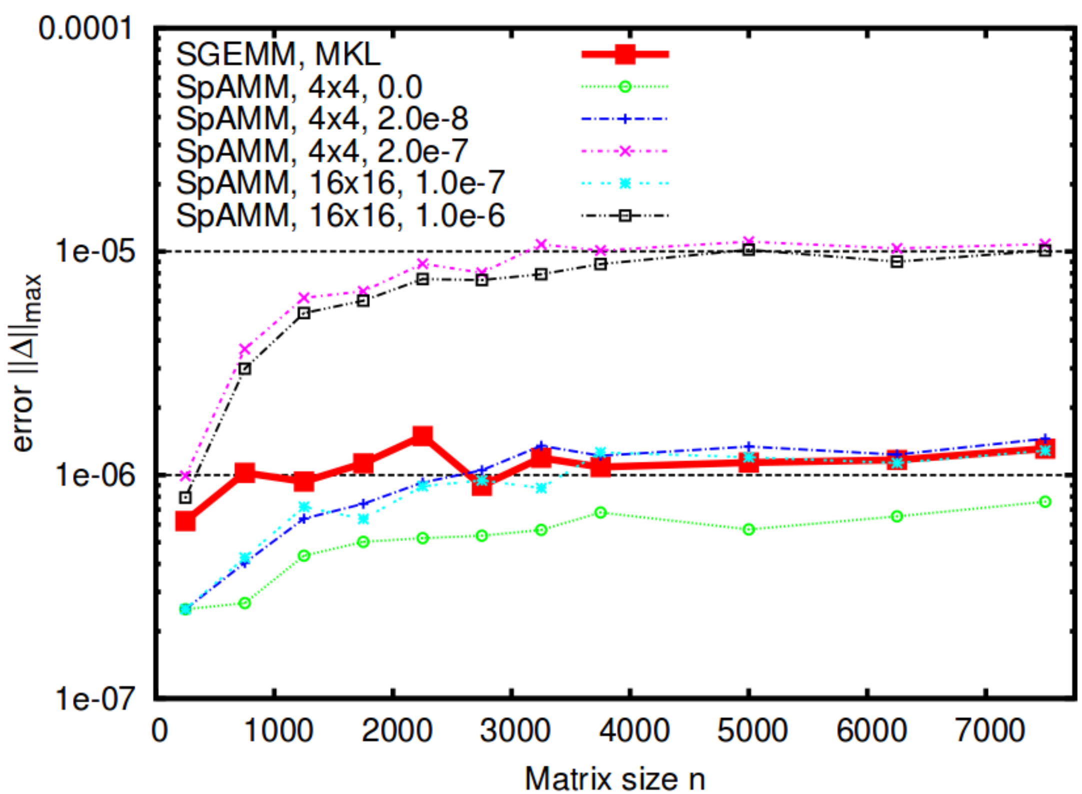


SpAMM For Dense Decay Matrices

Bock & Challacombe, SIAM J. Sci. Comput., 35(1), C72

Product matrix is asymptotically sparse, but only much, much later

Fill in of small blocks at negligible cost yields $O(N)$ scaling even for dense matrices



In single, SpAMM beats MKL SGEMM in error

Recursion w/locality more accurate than row-column

What About an Optimized SpAMM?

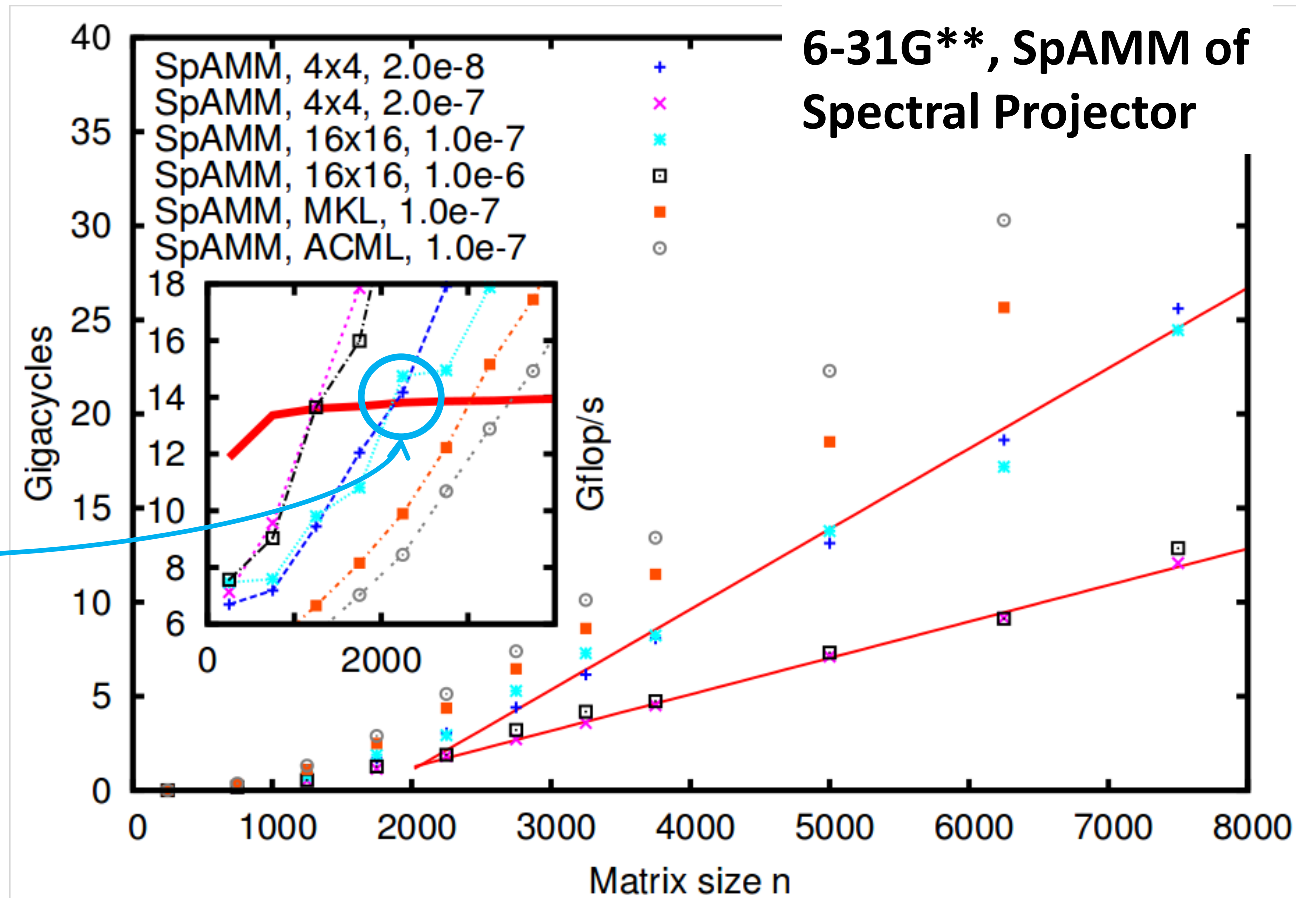
Bock & Challacombe, SIAM J. Sci. Comput., 35(1), C72

Assembly coded SpAMM in single. Don't try this at home…
Informs use of pragmas, OpenMP 4.0 directives & *etc.*

50% of peak with <u>4x4</u> blocking

Crossover with MKL/SGEMM at N=2000, <u>same error</u>

Can we do as well with directives (openmp 4)?



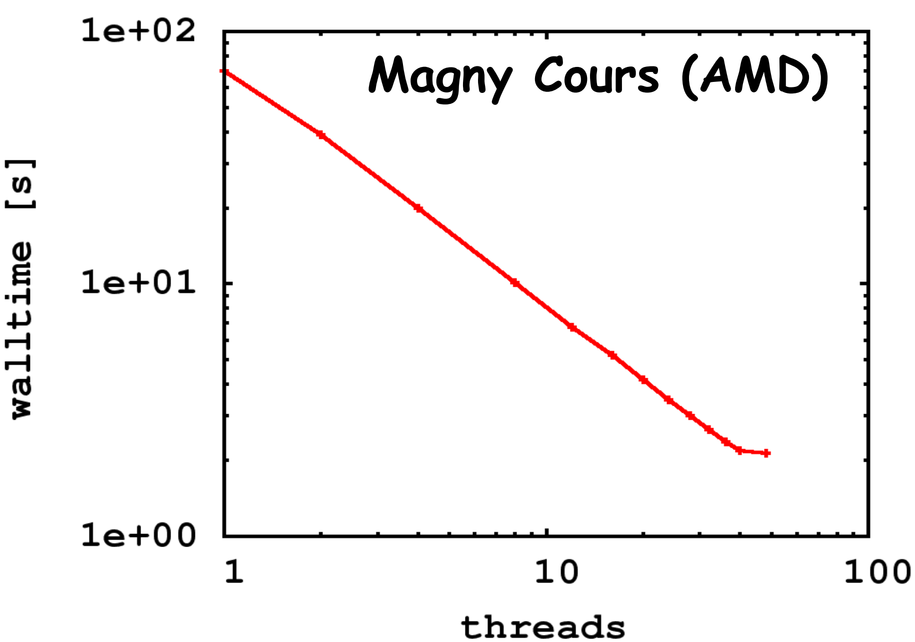
SpAMM – OpenMP 3.0

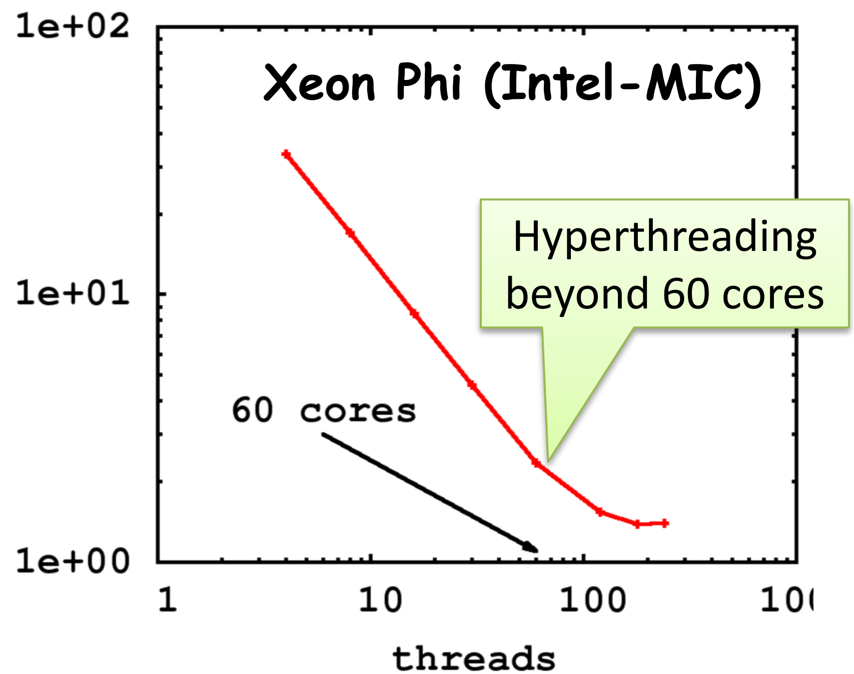
function MULTIPLY(τ, tier, A, B, C)
    if tier $<$ depth **then**
        for all $\{i, j, k \mid C_{ij} \leftarrow A_{ik} B_{kj}\}$ **do**
            if $\|A_{ik}\| \, \|B_{kj}\| > \tau$ **then** ▷ Eq. 1
                Create untied OpenMP task
                MULTIPLY(τ, tier+1, A_{ik}, B_{kj}, C_{ij})
            end if
        end for
        OpenMP `taskwait`
    else
        Acquire OpenMP lock on C
        $C \leftarrow C + A \times B$ ▷ Dense product, *e.g.* `BLAS`
        Release OpenMP lock on C
    end if
end function

SpAMM with OpenMP3 and OpenMP4

- OpenMP3 work very well with recursive task parallelism and atomic operations. Hardware support will improve performance.

- OpenMP4 may allow high performance for SIMD pipelines (ongoing). Our studies suggest 50% of peak possible even for very fine granularities (4x4).





Weak Scaling and the Parallel SpMM

Bowler *et. al*, arXiv:1402.6828 ← very entertaining comment …

So far, no code has demonstrated a parallel SpMM beyond ~ 4 atoms/core ($p < N/4$). While enabling bigger systems, does not enable high throughput (timesteps!)

Best results to date involve work homogenization and Cannon's method, precluding $p >> N$.

Argue that no SpMM employing matrix decomposition can be $O(N)$ & access strong scaling limit

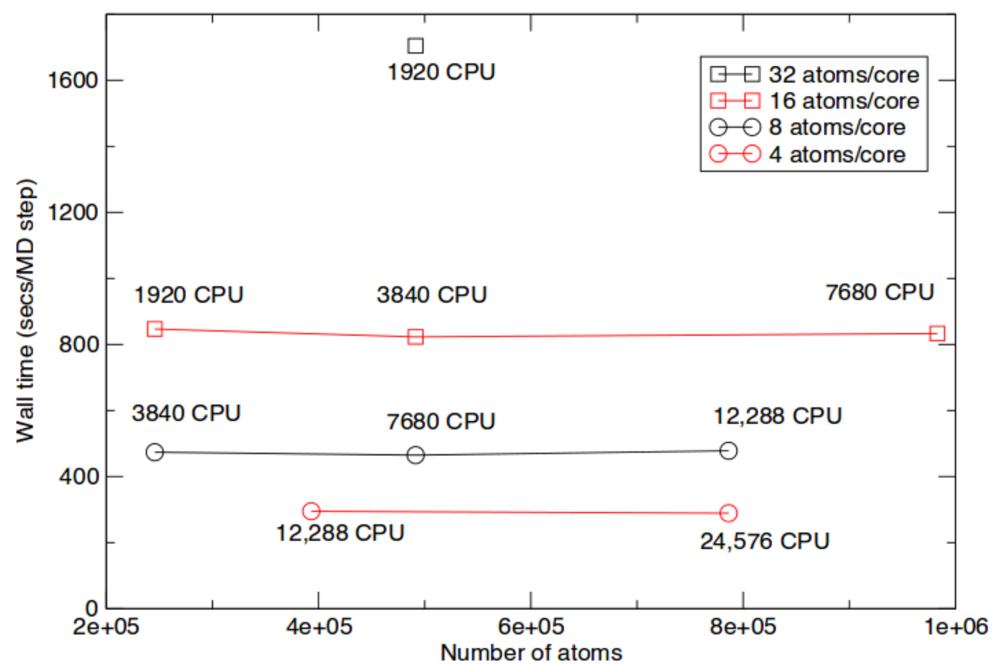


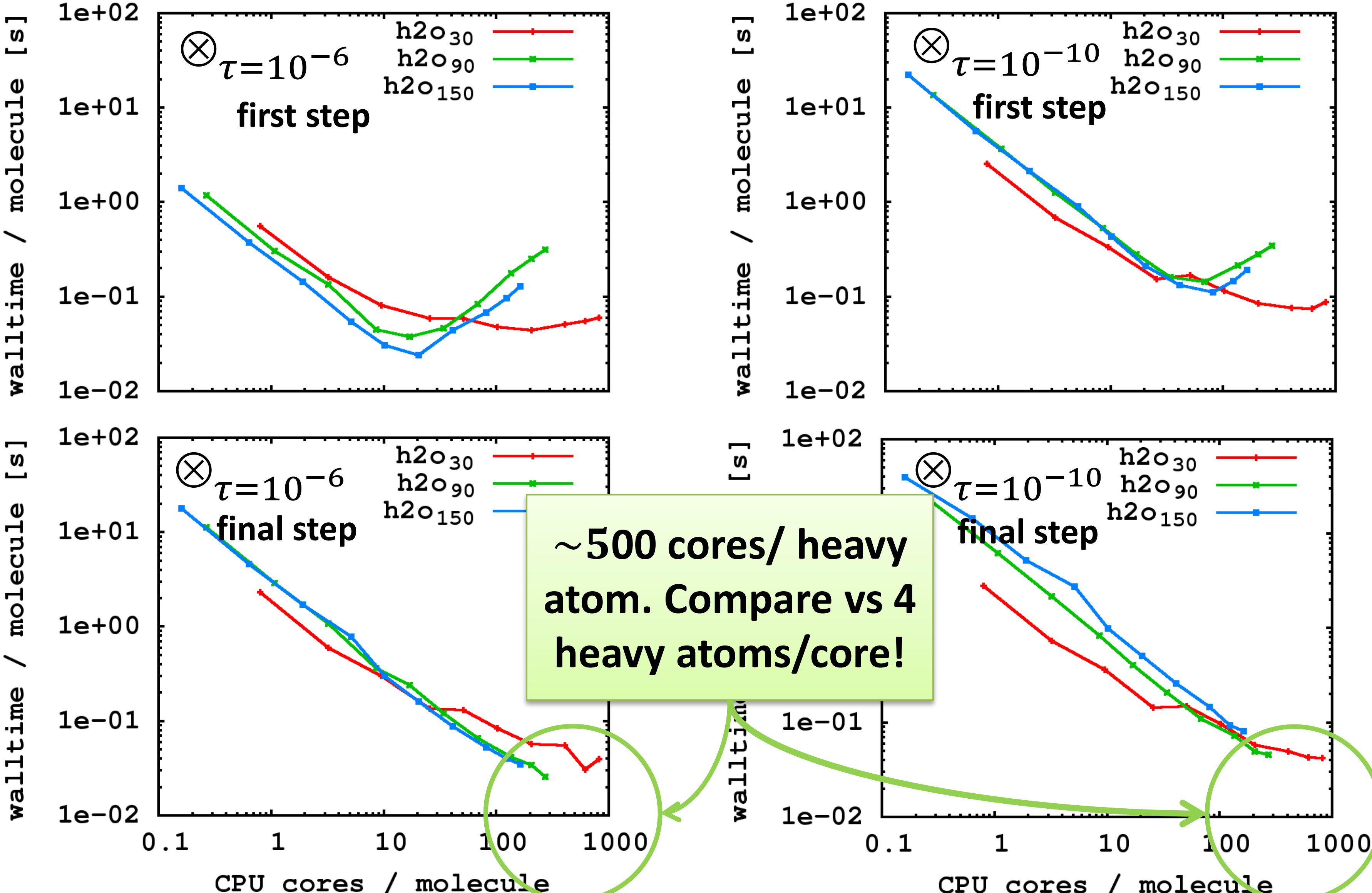
FIG. 1: Weak scaling for CONQUEST on the K computer, showing scaling to around 200,000 cores (8 cores per CPU).

SpAMM in the Strong Scaling Limit

Bock & Challacombe, SIAM J. Sci. Comput, to be accepted w/revs 2015

- ✓ Quantum locality → data locality via SFC heuristics
- ✓ **Decomposition in <u>3-D task space</u>, not in the sparse 2-D data space**
- ✓ Common runtimes that support recursive task parallelism (openmp/smp) and persistence based load balancing (charm++/distributed)
- Parallel scaling studies of functional approximation by iteration, for spectral projection (sign/step function).
    - first iteration is dynamically balanced, then with persistence load balancing after.

Dynamic and Persistence Load Balancing in SpAMM Spectral Projection (Sign Function)



∼500 cores/ heavy atom. Compare vs 4 heavy atoms/core!

Perfectly Strong Scaling & Communications Optimality

The Yelick and Demmel groups at Berkeley are hot on the trail of communication optimality for *n*-body interactions and perfect strong scaling for recursive matrix multiplication. Maybe we can do better still for SpAMM?

A Communication-Optimal N-Body Algorithm for Direct Interactions

Michael Driscoll*[1], Evangelos Georganas*[1], Penporn Koanantakool*[1], Edgar Solomonik*, and Katherine Yelick*[†]
*Computer Science Division, University of California, Berkeley
†Lawrence Berkeley National Laboratory, Berkeley, CA
{driscoll,egeor,penpornk,solomon,yelick}@cs.berkeley.edu

Strong Scaling of Matrix Multiplication Algorithms and Memory-Independent Communication Lower Bounds

Grey Ballard[*]
UC Berkeley
ballard@eecs.berkeley.edu

James Demmel[*†]
UC Berkeley
demmel@cs.berkeley.edu

Olga Holtz[‡]
UC Berkeley and TU Berlin
holtz@math.berkeley.edu

Benjamin Lipshitz[*]
UC Berkeley
lipshitz@berkeley.edu

Oded Schwartz[§]
UC Berkeley
odedsc@eecs.berkeley.edu

Metric Learning in Genomics

- Consider phylogenetic classification and metagenome resolution based on k-mer frequency vectors of dimension $n = 4^k$. K-mers are nucleotide strings of length k, widely available with modern technology.

- Define the generalized inner product between k-mer frequency vectors, $\langle \boldsymbol{x}, \boldsymbol{x}' \rangle_{\boldsymbol{M}} = \boldsymbol{x}^* \cdot \boldsymbol{M} \cdot \boldsymbol{x}'$. Then, the corresponding generalized distance is $\|\boldsymbol{x} - \boldsymbol{y}\|_M = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^* \cdot \boldsymbol{M} \cdot (\boldsymbol{x} - \boldsymbol{y})}$.

- **Objective**: Classification and resolution based on the learned metric $\boldsymbol{M}$ (novel).

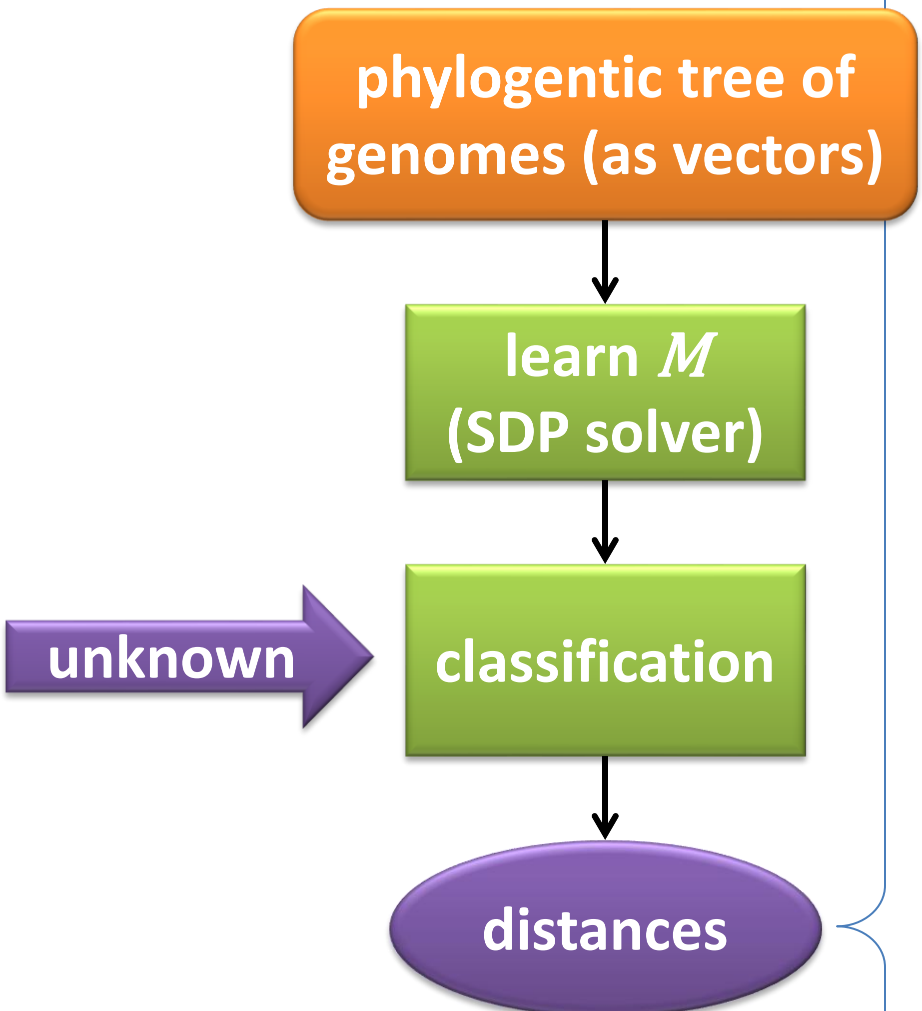
- **Challenge:** Exponential growth as $n = 4^k$, with unique sequences only starting to appear for $k > 11$.

Learning the Mahalanobis Metric with Convex Optimization

$$\min_{\boldsymbol{M}} \left[\underbrace{\sum_{x,y \in \mathcal{S}} \langle x - y, x - y \rangle_M}_{\text{Minimize distances between similar vectors } (\mathcal{S})} - \underbrace{\sum_{x,y \in \mathcal{D}} \langle x - y, x - y \rangle_M}_{\text{Maximize distances between dissimilar vectors } (\mathcal{D})} \right] \underbrace{\text{s.t. } \boldsymbol{M} \succcurlyeq 0 \ \& \ \mathrm{tr}\boldsymbol{M} = 1}_{\text{Maintain } \boldsymbol{M} \text{ Hermetian, positive semidefinite, and scale invariant}}$$

- Can solve this problem with Semi-Definite Programming (SDP) in $O\left(m^2 n^2 + \boldsymbol{n^3}\right)$, where *m* is the training set size, and *n* is the matrix (learning) dimension.
- **Reduce the cost from $\boldsymbol{O\left(n^3\right)}$ to $\boldsymbol{O(n)}$ with SpAMM and gradients only approach to SDP.**

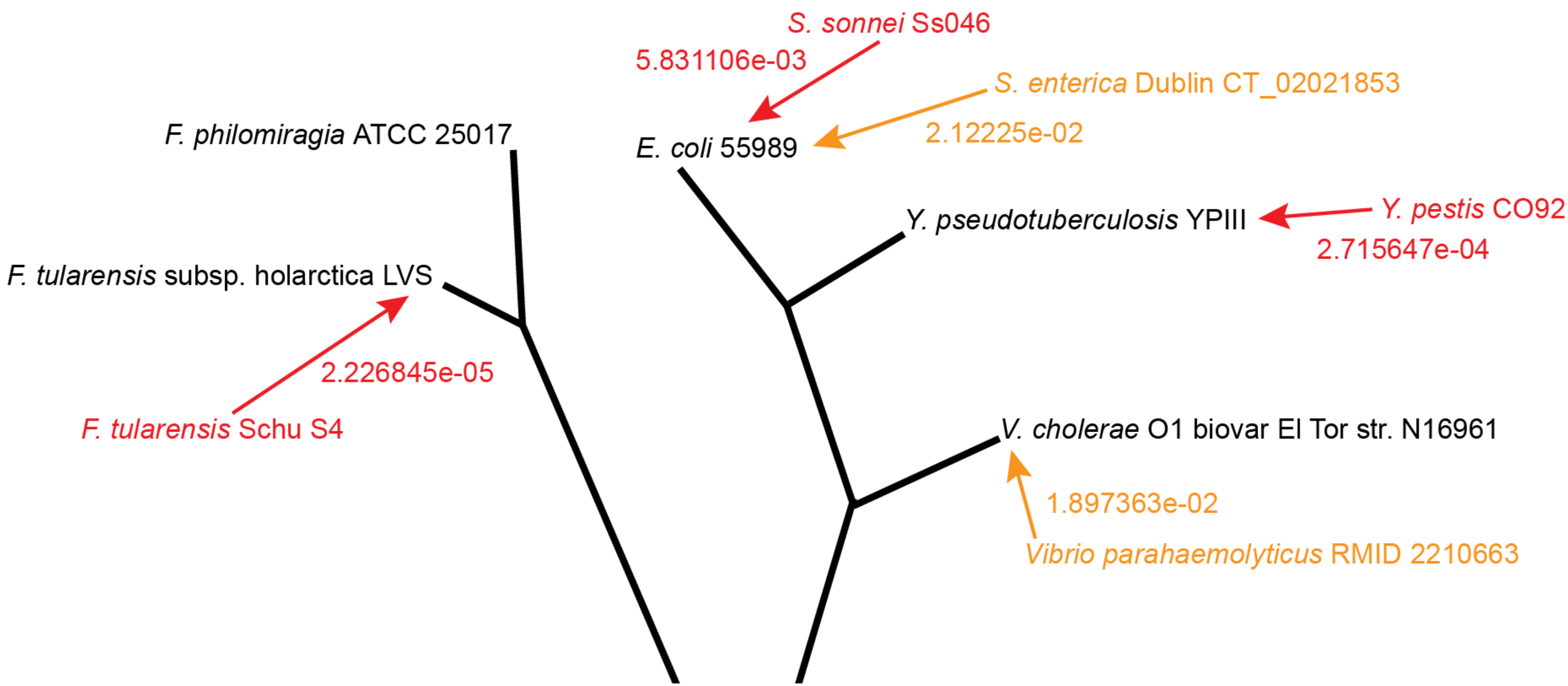
Metric Classification of Unknowns with k=5



Faux Unknown	Closest training set genomes	Distance
F. tularensis subsp. tularensis FSC198	Francisella tularensis subsp. tularensis Schu S4	5.17e-11
	Francisella tularensis subsp. mediasiatica FSC147	2.69e-08
	Francisella tularensis subsp. tularensis WY96-3418	2.75e-08
	Francisella tularensis subsp. holarctica FTNF002-00	7.01e-08
	Francisella tularensis subsp. holarctica OSU18	1.12e-07
	Francisella tularensis subsp. holarctica LVS	1.29e-07
	Francisella novicida U112	2.44e-07
	Francisella philomiragia ATCC 25017	1.51e-05

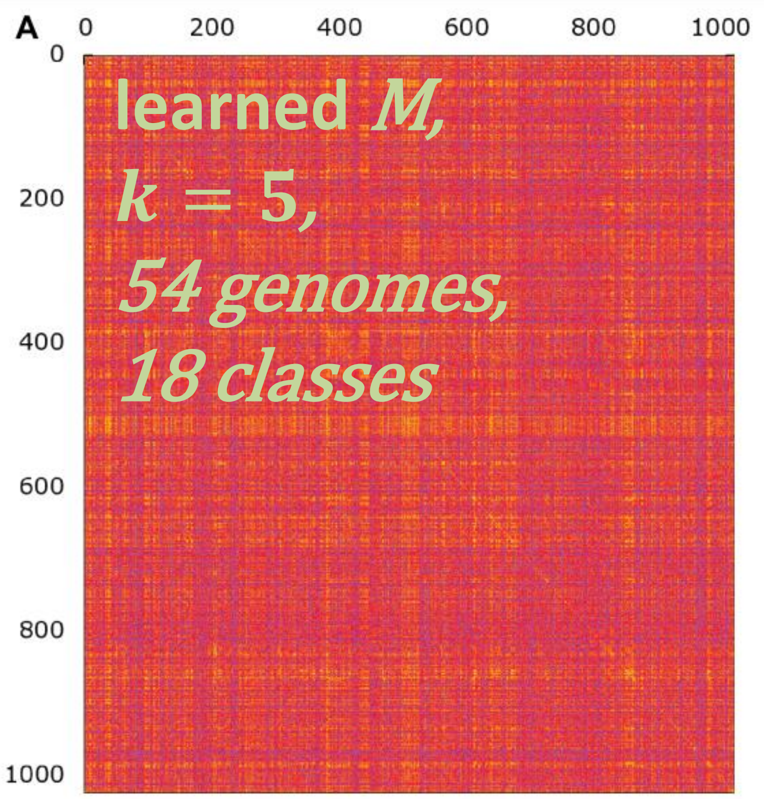
Metric Classification of Unknowns with *k*=5

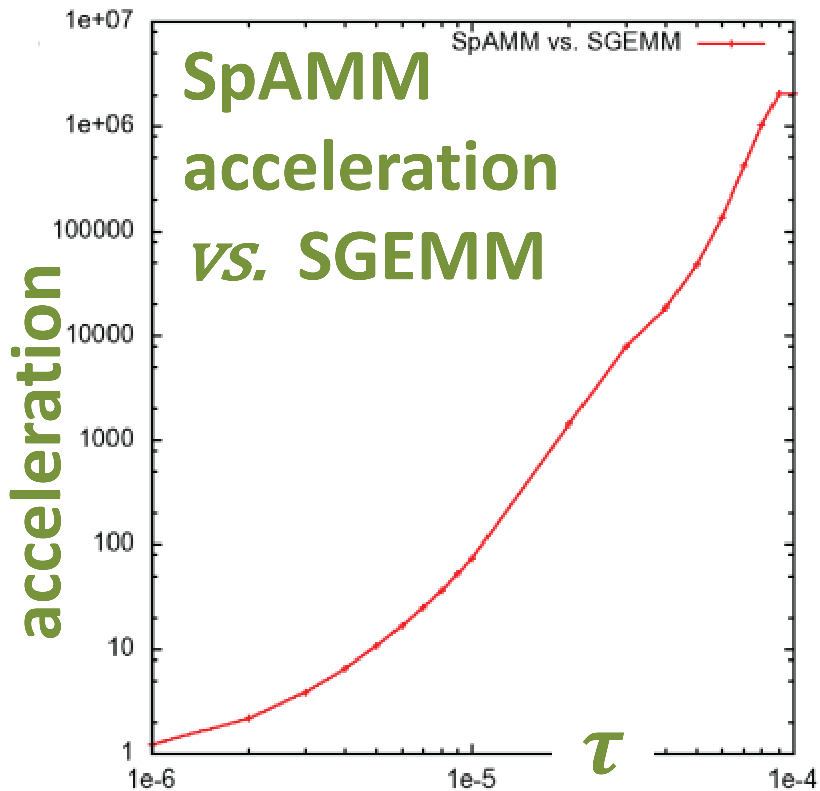
- Test: train with just 5 genomes (simple but hard)
  - ✓ correctly classify 5 fake unknowns
- Get better resolution with larger training sets ***and larger k***



SpAMM & Gradients only SDP for $k>>5$?

- To maintain $\boldsymbol{M} \succcurlyeq 0$ with gradients only, need to solve the line search problem: $\min_{\alpha} f(\boldsymbol{P}_{+}[\boldsymbol{M} + \alpha \nabla f(\boldsymbol{M})])$. Use fast methods to find the spectral projector $\boldsymbol{P}_{+}[\boldsymbol{M}]$. Works crudely.
- $\boldsymbol{M}$ is a correlation matrix, so expect (and find) decay properties. Better locality heuristics for k-mers will enhance SpAMM acceleration





Solver Ecosystems Towards Materials Design

PHYSICS: **Strong Correlation**

- single determinant KSTs. Correlation on top of ***Fock exchange***, eg. B13.
- toward Mott transition, ***ill-conditioned matrix functions***

MATH: **Functional Approximation**

- nested approximate algebras and recursive preconditioning.
- ***ill-conditioned matrix functions***

N-BODY SOLVERS: **Generalized Thin Stacks**

- *N*-Body Fock exchange (NoFX)
- *N*-Body Linear Algebra (SpAMM)
- many others …

InfoSci: ***N*-Body Learning**

- fast kernel summation.
- metric learning: fast approaches to linear algebra including semi definite programming.

CompSci: **Generic Programming**

- functional programming, skeletons, recursive task parallelism, openmp 4.
- enterprise frameworks: scala/spark + neo/epiphany/phi.

Enterprise Trends for Solver Ecosystems

- Towards generic and communication optimal solver collectives
- Leverage disruptive technologies along the commodity trend: enterprise frameworks like spark, and the cost deflation of decentralized resources like EC2 cloud.

thin, generalized stacks for *n*-body collectives → distributed task and vector parallelism under openmp 4.0 → decentralized, generic parallelism, via *eg.* scala/spark

$$O(N^x) \rightarrow O(N) \rightarrow \frac{5\mathrm{GF}}{\mathrm{Watt}} O(N/p) \rightarrow \frac{100\mathrm{GF}}{\mathrm{Watt}} O(N/p)$$