Towards Fast Scalable Solvers for Charge Equilibration in Molecular Dynamics Applications

Kurt A. O'Hearn
H. Metin Aktulga

Department of Computer Science and Engineering
Michigan State University

ScalA16: Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems
2016-11-13

Towards Fast Scalable Solvers for Charge Equilibration in Molecular Dynamics Applications

Background

- Domain: atomic-level molecular dynamics
  - Classical MD: good scalable methods, but static bonding and fixed charges limit modeling capability
  - Reactive, polarizable MD: expand modeling ability but pose new computational challenges: dynamic charge distribution for reactivity
    - Idea: distribute charge (electrons) to atoms according to min. energy configuration (ground state)
- Goal: improve performance of expensive, poorly scaling charge distribution kernel
- Models for Charge Distribution:
  - charge equilibration (QEq)
  - electronegativity equilibration (EEM)
  - atom condensed Kohn-Sham DFT approximated to second order (ACKS2)
  - split charge equilibration (SQE)
  - others (active research)
- Approaches for Finding Charges in Models:
  - linear solvers: iterative (GMRES with preconditioning)
  - extended Lagrangian

Towards Fast Scalable Solvers for Charge Equilibration in Molecular Dynamics Applications

QEq Overview

$$\begin{aligned}
\text{Minimize } \mathbf{E_{ele}}(\mathbf{Q}) &= \sum_i \chi_i q_i + \frac{1}{2}\sum_{i,j} H_{ij} q_i q_j \\
\text{where } H_{ij} &= J_i \delta_{ij} + \frac{1-\delta_{ij}}{\sqrt[3]{r_{ij}^3 + \gamma_{ij}^{-3}}} \\
\text{subject to } Q_{net} &= \sum_i q_i
\end{aligned}$$

After applying the Lagrange multiplier method:

$$\begin{aligned}
-\chi_k &= \sum_i H_{ki} s_i, \qquad k = 1, \ldots, n \\
-1 &= \sum_i H_{ki} t_i, \qquad k = 1, \ldots, n
\end{aligned}$$

where:

$$q_i = s_i - \frac{\sum_i s_i}{\sum_i t_i} \cdot t_i$$

- $\mathbf{R} = (r_1, r_2, \ldots, r_n)$: positions in system with n atoms ($\in \mathbb{R}^3$)
- $\mathbf{Q} = (q_1, q_2, \ldots, q_n)$: partial charges (unknowns)
- χ_i: electronegativity of atom i
- J_i: idempotential of atom i
- $r_{ij} = ||r_j - r_i||_2$: distance between atoms i and j
- γ: electrostatic shielding term

Preconditioner Design and Considerations

Design

- Idea: use good initial guesses (already done via extrapolation)
- Idea: apply preconditioning to GMRES residual vector
  - Relatively different than those for PDE discretizations: much higher NNZ per row on average

Considerations

- Effectiveness: reduction in num. of solver iters.
- Cost: computation and application time
- Longevity: num. of sim. iters. for which preconditioner is usable
- Parallelizabilty: shared mem. scalability of preconditioner
  - Important building block for dist. mem. algorithms!

Preconditioner Computation Methods

Basic Techniques

- Diagonal inverse (**Diag**)

$$\mathbf{P}^{-1}_{ij} = \begin{cases} \frac{1}{\mathbf{H}_{ij}}, & i = j \\ 0, & \text{otherwise} \end{cases}$$

- Incomplete factorizations with 0 fill-in
  - Cholesky with level scheduling (**IC(0)**)
    - Idea: analyze sparsity pattern, perform parallel operations on independent levels

$$\mathbf{H} \approx \mathbf{U^T U}$$

  - Fine grained (**FG-ILU(0)**)
    - Idea: asynchronous constraint-based approach

$$\mathbf{H} \approx \mathbf{LU}$$

Preconditioner	**Cost**	**Longevity**	**Effectiveness**	**Parallelism**
Diagonal	Very Low	High	Low	High
IC(0)	High	Moderate	High	Low
FG-ILU(0)	Very High	Moderate	High	High

Preconditioner Computation Methods

Enhanced Techniques

- Distance drop (**IC-dist(d)**, **FG-ILU-dist(d)**)
  - Insight: off-diagonal entries are inversely proportional to distance between atoms

$$r_{ij} > r_{\text{nonb}} \cdot d, \qquad d \in (0, 1)$$

- Dual drop (**IC-dual(t)**, **FG-ILU-dual(t)**)
  - Numeric thresholding based on row-wise 1-norm

$$H_{ij} > ||H_{i,:}||_1 \cdot t, \qquad t \in \mathbf{R}^+$$

  - Apply during and post factor computation for **IC-dual(t)**, but only post for **FG-ILU-dual(t)** to achieve convergence

Preconditioner Application Methods

Techniques for Incomplete Factorizations

- Level scheduling
  - Symbolic stage: find available parallelism in triangular solves (once per preconditioner computation)
  - Numeric stage: perform solves in parallel per level
  - Combine with **IC(0)**, **IC-∗(∗)** computation methods
- Jacobi iteration
  - Approximate traingular solve:
    - To solve the triangular system $Fx = b$, instead iterate

$$x_{k+1} = \left(I - D^{-1}F\right) x_k + D^{-1}b$$

  - Fixed num. of iters. (SpMV, vector-vector ops.)
  - Combine with **FG-ILU(0)**, **FG-ILU-∗(∗)** computation methods

Numeric Experiments

Setup

- Platform
  - NERSC Edison: Cray XC30
    - 12-core Intel Ivy Bridge, 2.4 GHz (dual socket)
    - Private L1 (32 KB inst., 32 KB data), L2 (256 KB) caches
    - Shared 30 MB L3
    - 64 GB DDR3, 1866 MHz bus
- Software and Tools
  - PuReMD ReaxFF
  - Intel compiler v15.0.1
  - Optimization flags: `-fast`
- Chemical Systems

Name	**Chemical Formula**	**#Atoms**	**Type**
Water	H_2O	78,480	Liquid
Silica	SiO_2	72,000	Amorphous material
PETN	$C_5H_8N_4O_{12}$	48,256	Perfect crystal
Bilayer	Phospholipid bilayer	56,800	Soft matter

Numeric Experiments

Cost-Effectiveness Trade-off

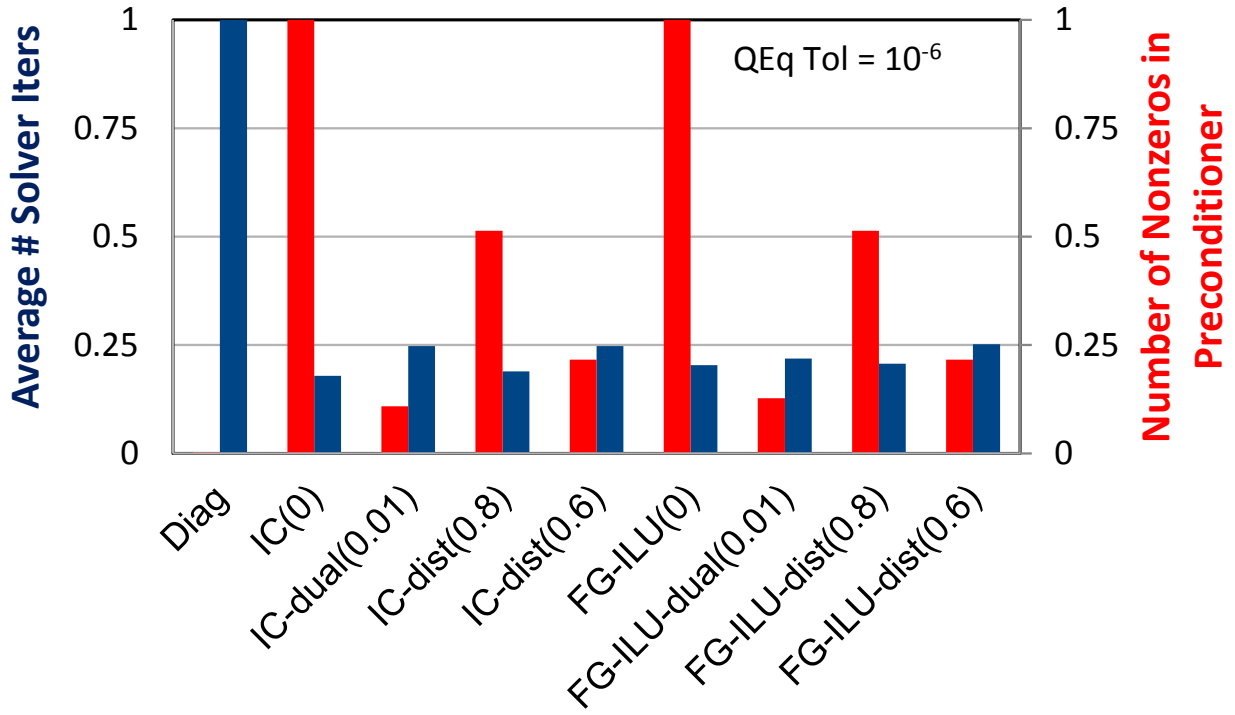

Figure: Effectiveness vs. expected cost of various preconditioned GMRES solvers (averaged across all four benchmark systems)

Numeric Experiments

Longevity

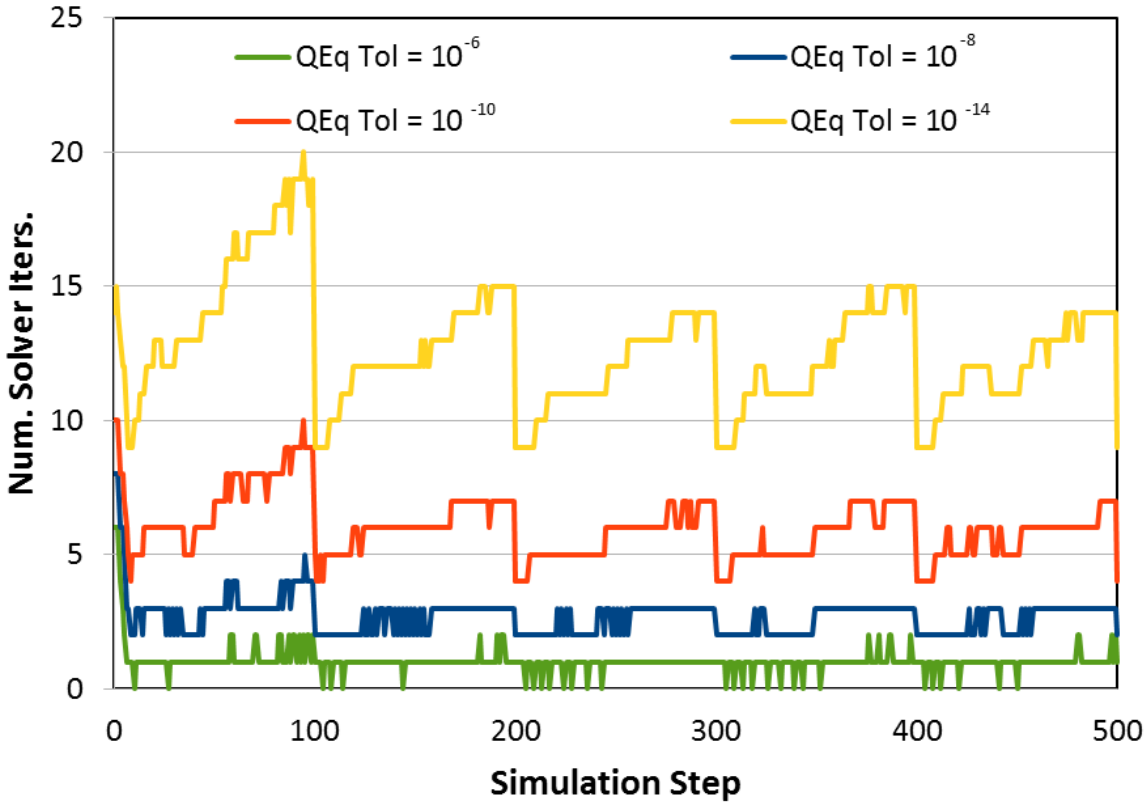

Figure: **FG-ILU-dual(0.01)** for the bulk water system, 100 steps before re-computing preconditioner

Numeric Experiments

Overall Solver Performance

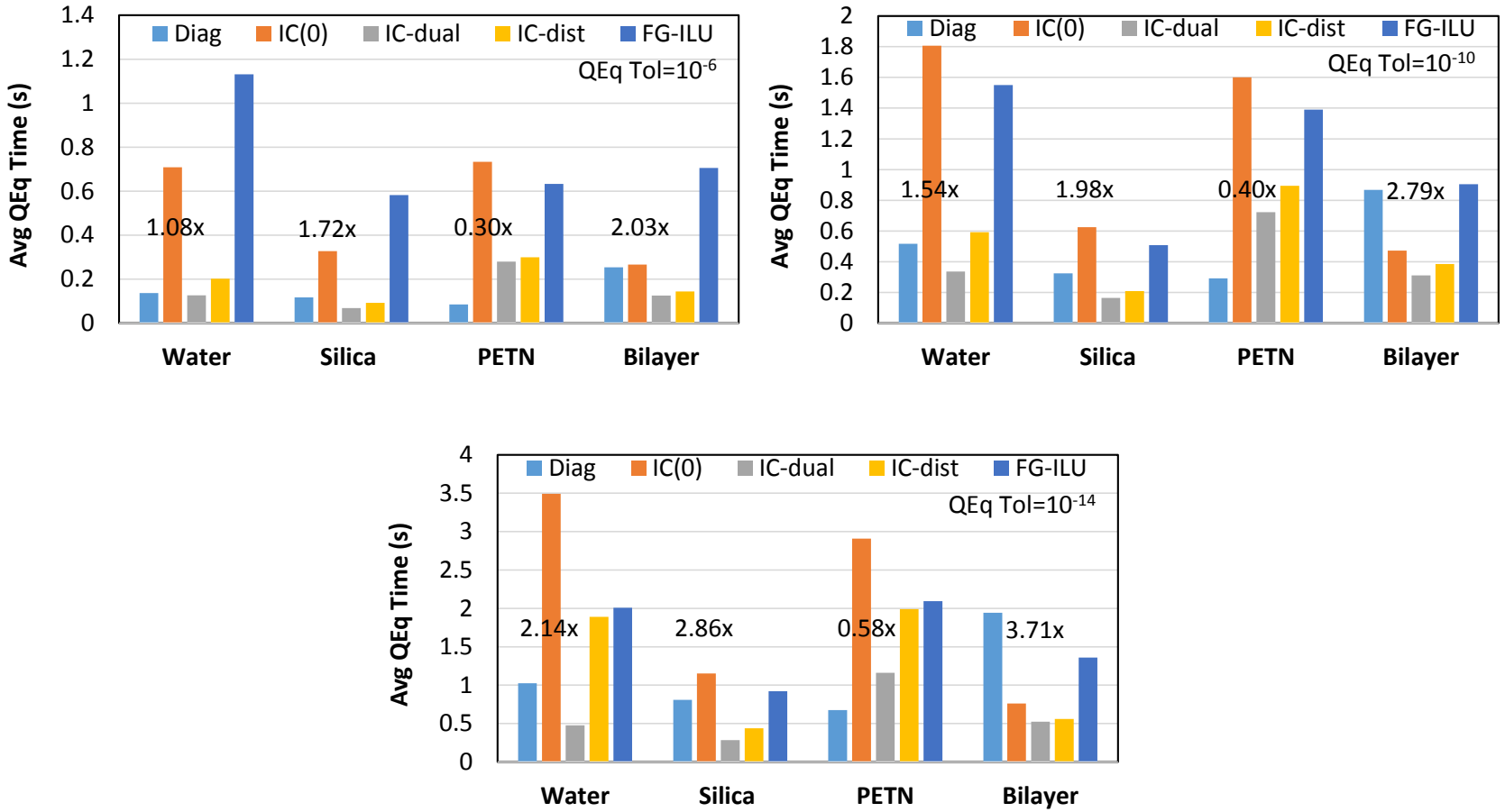

Figure: Average QEq times for different solvers, numbers over each bar group show the speedup achieved by the IC-dual scheme over the Diag solver

Numeric Experiments

Thread Scalability

Figure: **IC-dual(0.01)** with water (left) and PETN (right) at solver tol. 10^{-14}

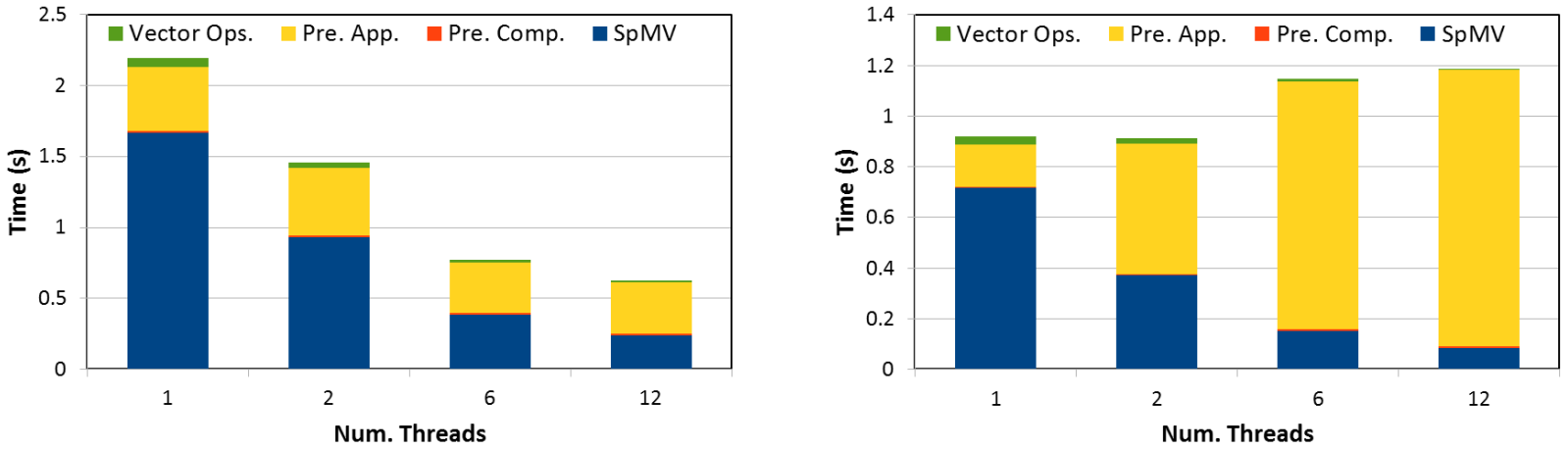

Table: Num. of levels in preconditioner factors

System	IC(0)	IC-dual(0.01)	IC-dist(0.8)	IC-dist(0.6)
Bilayer	4921	711	3280	1851
PETN	13181	13181	39078	16003
Silica	10799	634	6147	1705
Water	25479	1769	12365	5142

Future Work and Acknowledgements

Future Work

- Reduce num. levels and sync. of PreApp through explicit permutations via graph coloring
- Experiment with further combinations of enhanced preconditioning techniques
- Implement approaches on many-core architectures (Xeon Phi)
- Develop an automated parameter tuning method

Acknowledgements

- MSU Foundation Strategic Partnership Grant
- NSF Grant ACI-1566049
- NERSC under Contract No. DE-AC02-05CH11231