Ordering Schemes for Preconditioning Sparse Incomplete Factors

Ingyu Lee and Padma Raghavan
Department of Computer Science and Engineering
The Pennsylvania State University

Esmond G. Ng
National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory

Introduction

- We consider the solution $Ax = b$ using preconditioned CG (A sparse, SPD)
- Incomplete factorization preconditioners discard some or all the *fill* in the true Cholesky factor L ($A = LL^T$) to obtain the preconditioner $\hat{L}$
- Such schemes inherit many features of sparse factorization including the need to limit fill (original zeroes that become nonzero)
- We consider the role of orderings that specifically limit fill in $\hat{L}$ (as opposed to orderings that limit fill in L)

Background

- Consider computing L where $A = LL^T$:
  - The number of non-zeros in L ($|L|$) depends on a permutation of A
  - A permutation P is applied to A before computing $PAP^T = LL^T$
  - P is computed using greedy or divide and conquer methods
- Consider computing the incomplete factor $\hat{L}$:
  - One option is to use P selected to reduce fill in $PAP^T = LL^T$; $\hat{L}$ is then computed by discarding some/all fill in L
  - Alternatively, ordering schemes (not necessarily suitable for Cholesky factorization) can be developed to improve the performance of $\hat{L}$ as a preconditioner

Types of Incomplete Factorization

- Level of Fill (k):
  - $L_{i,j} \neq 0$ iff there is a $i - j$ 'fill-path' in $G(A)$, i.e., a path connecting vertices i and j in which all intermediate vertices are numbered less than $\min(i, j)$
  - $\hat{L}$ using k levels of fill: $\hat{L}_{i,j} \neq 0$ is there is an $i - j$ fill-path of length $\leq (k + 1)$
  - Final nonzero structure of $\hat{L}$ is known after ordering
- Drop Threshold:
  - $\hat{L}i, j = 0$ if $\frac{|\hat{L}i,j|}{\hat{L}_{j,j}} <$ threshold, dropped values are not used
  - Final nonzero structure $\hat{L}$ is known only at factorization

Orderings to Compute $\hat{L}$

- For $\hat{L}$ computed with orderings for L:
  - RCM (envelope reduction) schemes for very sparse $\hat{L}$ lead to better preconditioners
  - When more non-zeroes in $\hat{L}$ are needed, MMD orderings are quite competitive with RCM
  - Convergence with nested dissection orderings seem to lag those from RCM/MMD
- Specific orderings for $\hat{L}$:
  - Schemes such as 'Minimum Discarded Fill' have been developed that use metrics related to the norm of the discarded fill
  - These have shown improved quality of preconditioning but are typically expensive to compute

Our Approach: ICMD

- We consider orderings schemes that primarily use structural (symbolic) metrics to explicitly reduce fill in $\hat{L}$ (and not in L)
- The ordering is interleaved with numeric factorization to compute $\hat{L}$
- We consider greedy 'minimum degree' schemes based on the elimination graphs of $\hat{L}$

ICMD Algorithm

1. Initialize $\hat{L}$ to A
2. For $i = 1, 2, \ldots, n$, do
   (a) Select a column with minimum degree in $G(\hat{L})$; number it i.
   (b) Factor column $\hat{L}_{*i}$
   (c) Retain/drop $\hat{L}_{ji}$ using levels of fill/threshold
   (d) Update $\hat{L}_{*k}$ if $\hat{L}_{ki} \neq 0$, $\hat{L}_{*k}$ not factored

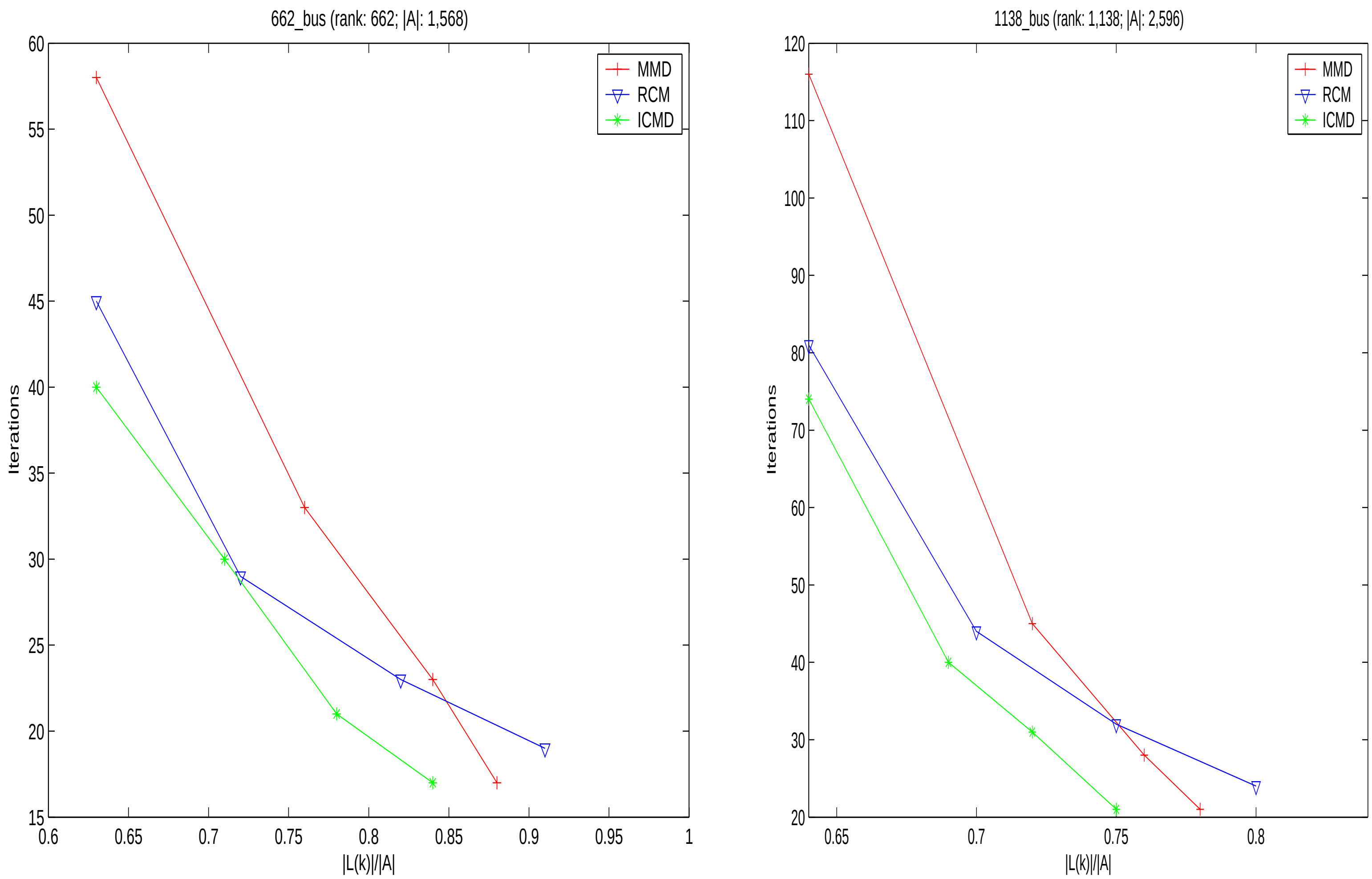
ICMD Implementation

- Ordering and computing $\hat{L}$ are done at the same time
- Graph-theoretic ideas cannot be easily used as in the MMD implementation especially for high levels of fill
- Drop-threshold schemes make it simpler to use cliques/and other compact structures to manage degree update
- Issues such as mass and multiple elimination have not yet been explored
- Tie breaking using numeric measures can be combined with the degree metric for reducing fill in $\hat{L}$

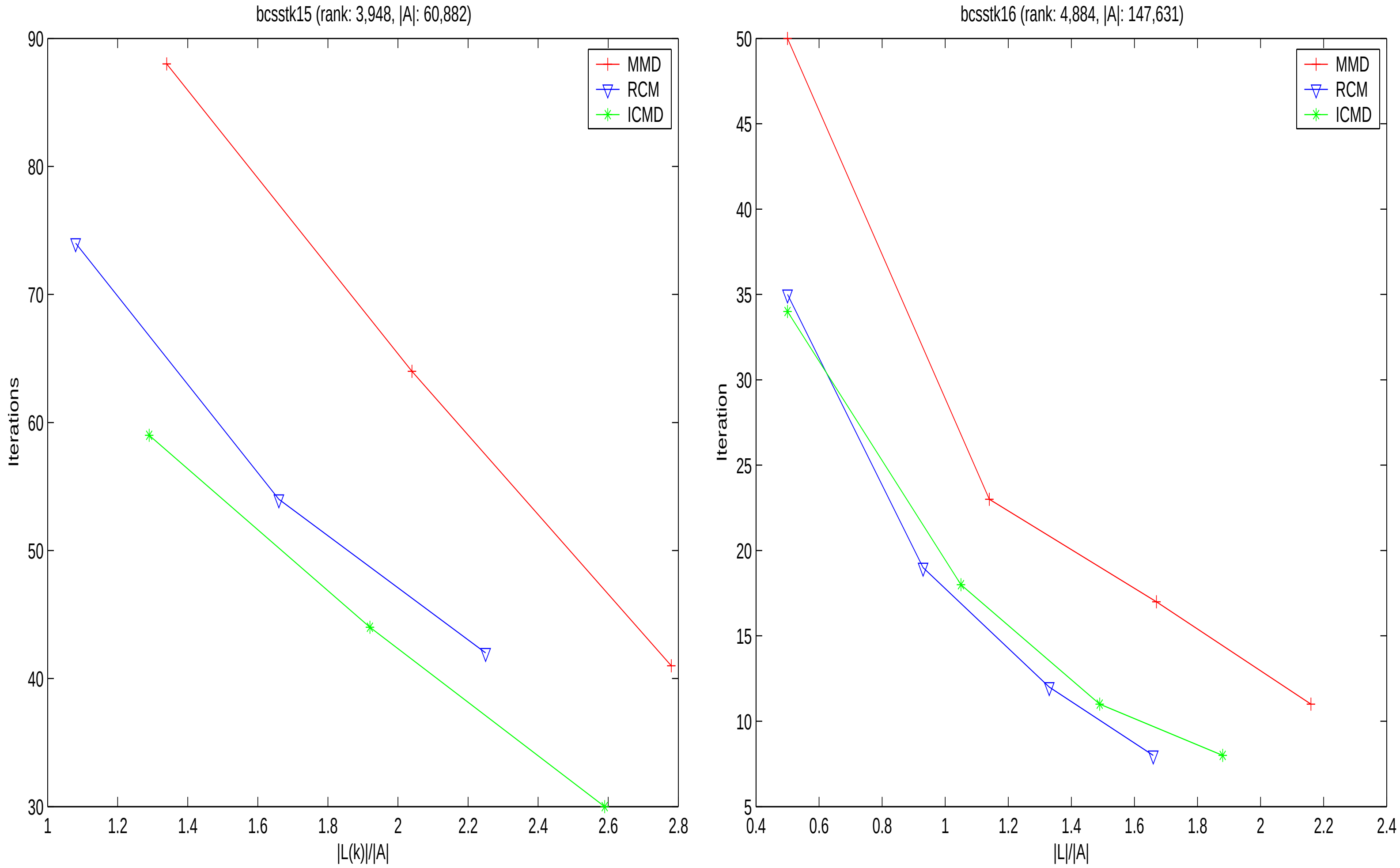
ICMD: Empirical Results

- We experimented with ICCG using ICMD with both levels of fill and drop-threshold ($\approx$ 100 instances)
- We show sample plots of iteration counts (Y-axis) againt fill along X-axis.
- Improvements are more noticeable for very sparse $\hat{L}$

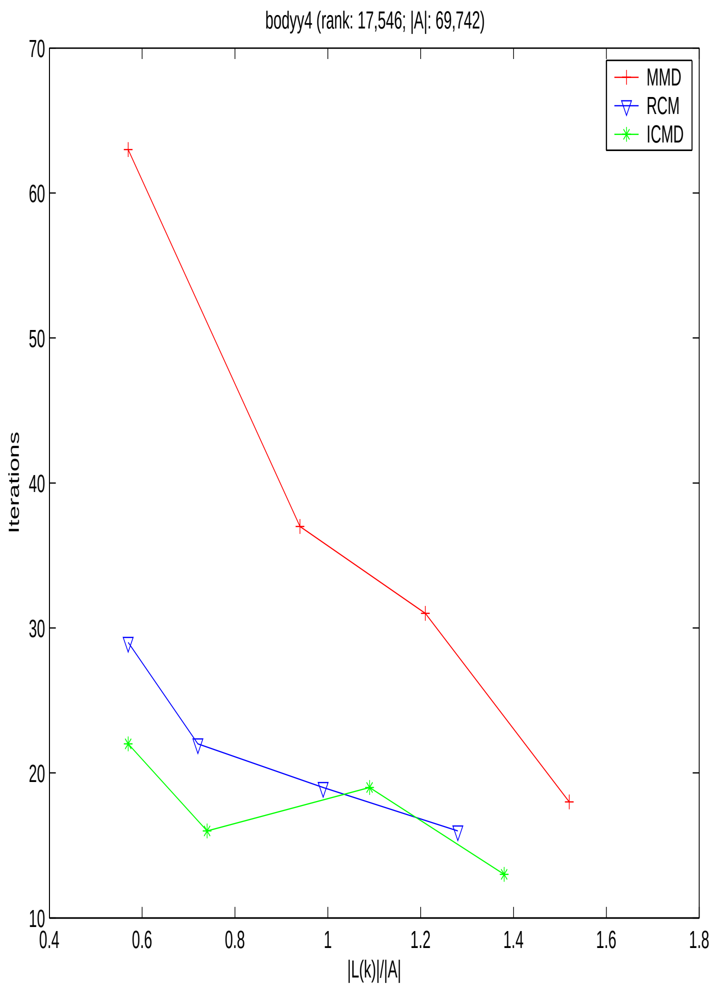
Level of Fill IC: bus matrices

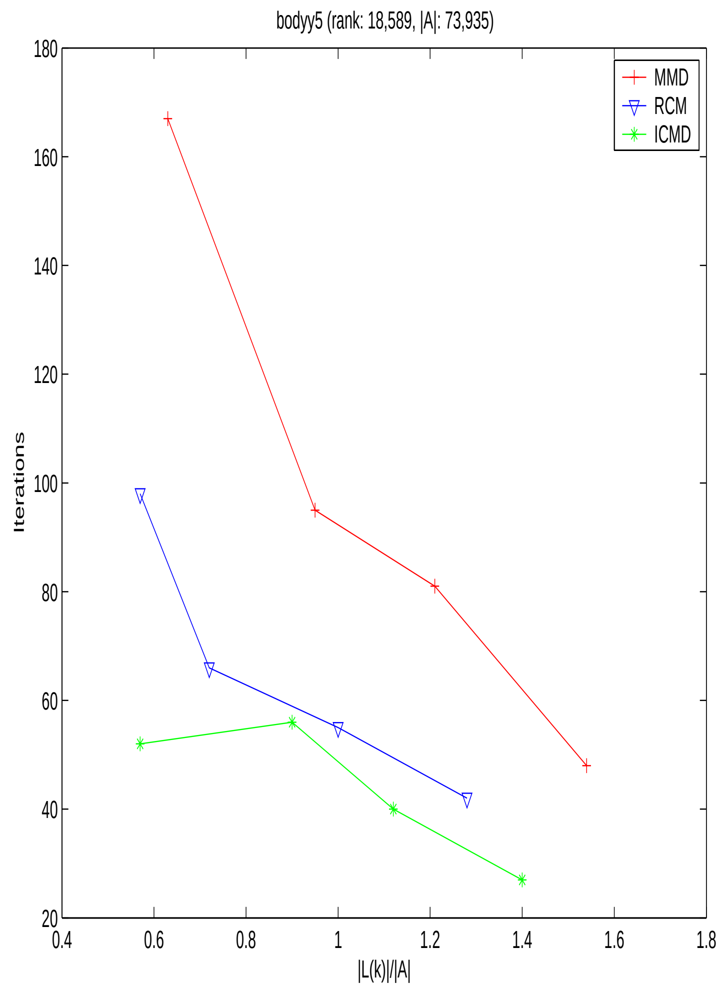


Level of Fill IC: bcsstk matrices

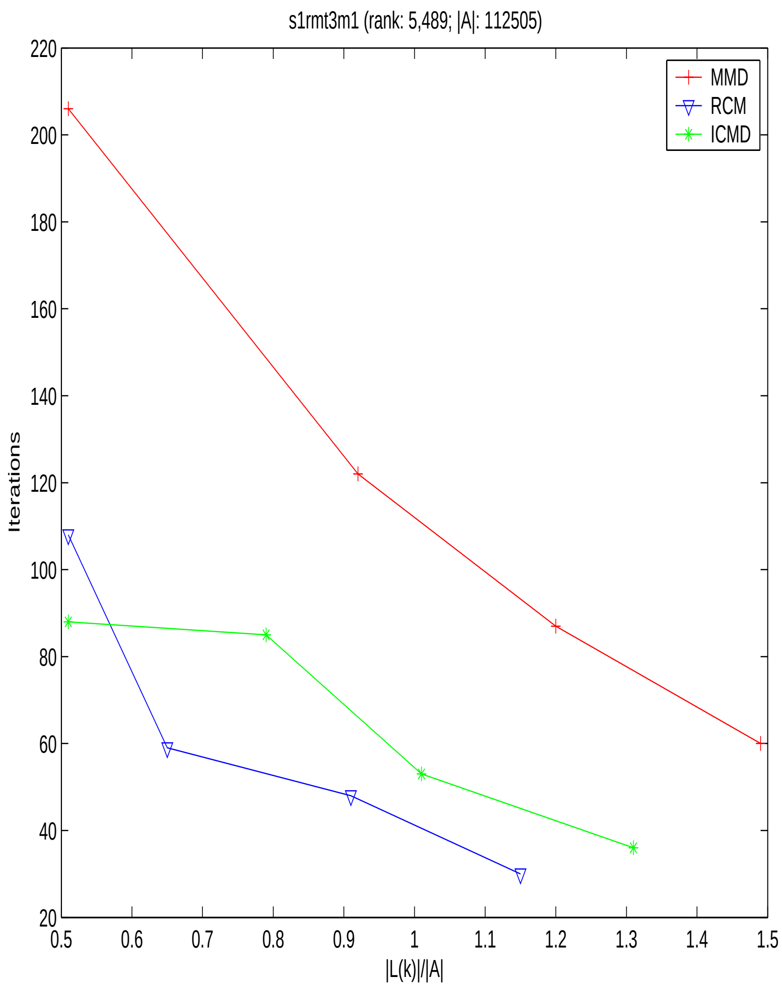


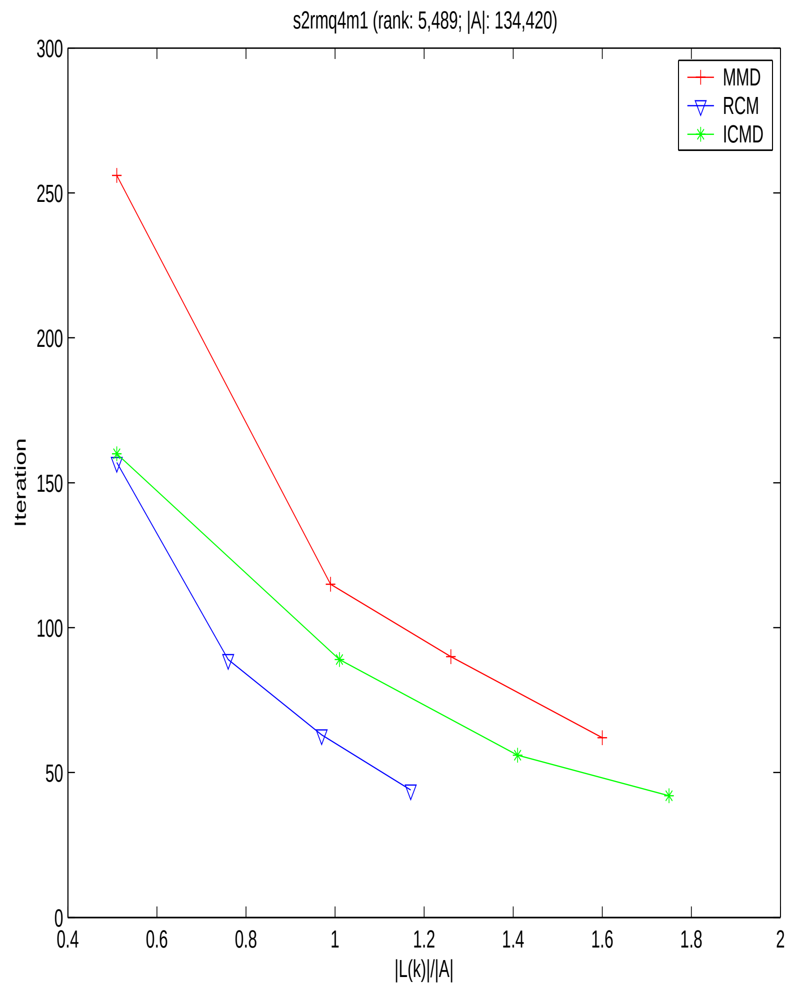
Level of Fill IC: bodyy matrices



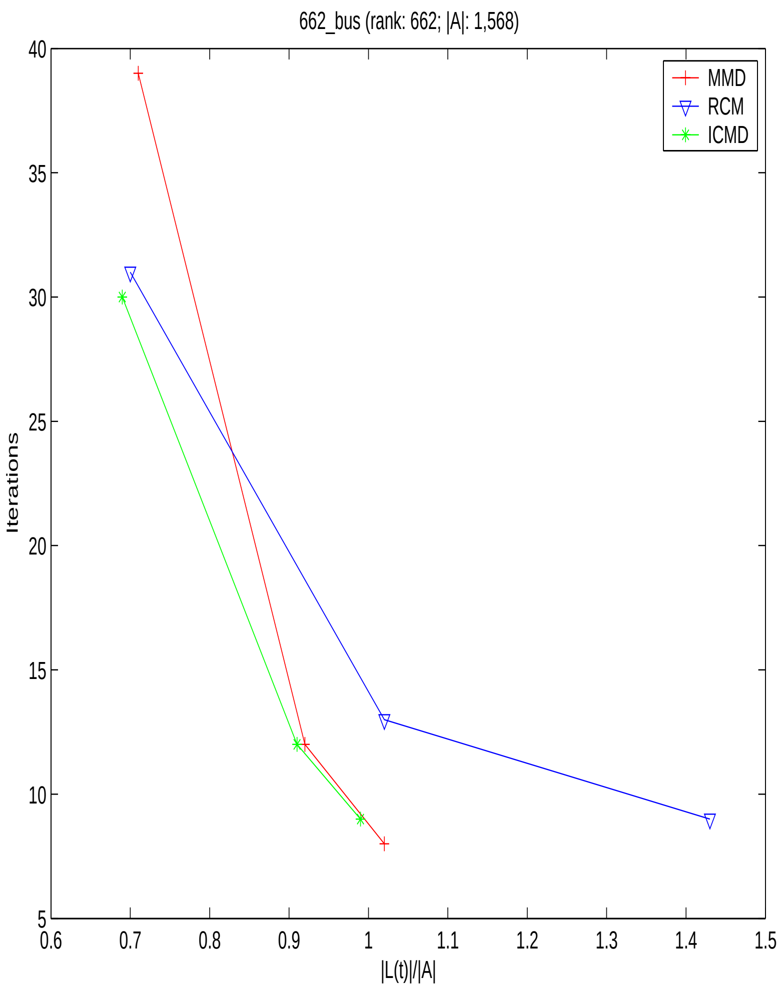


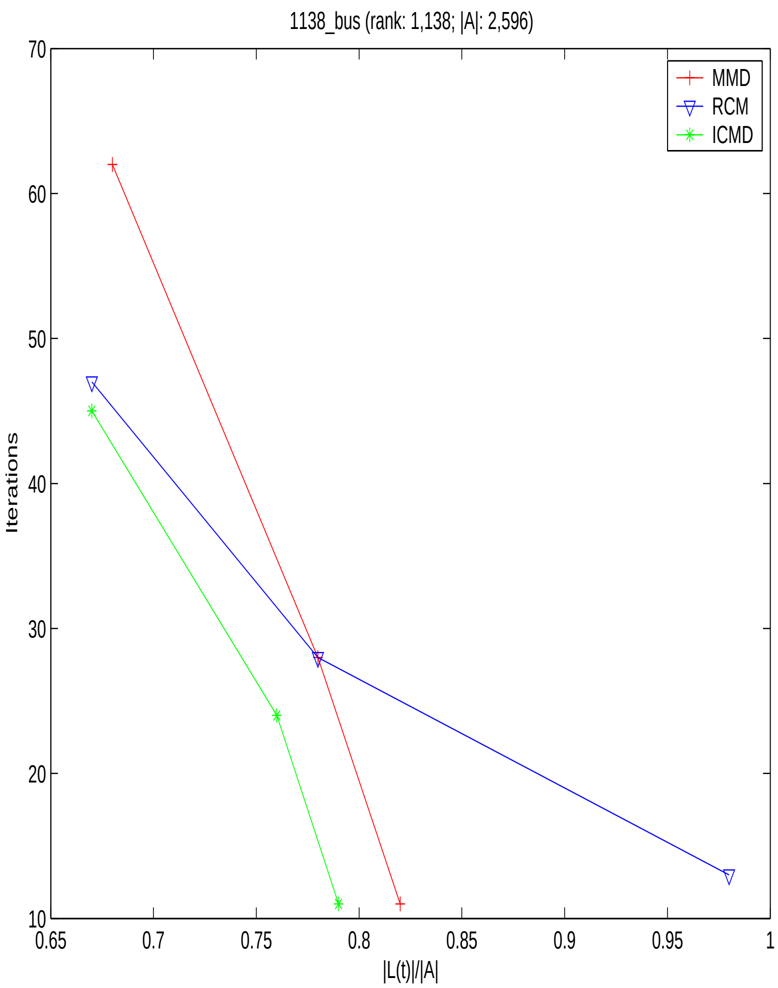
Level of Fill IC: `s?rmt?q/m` matrices



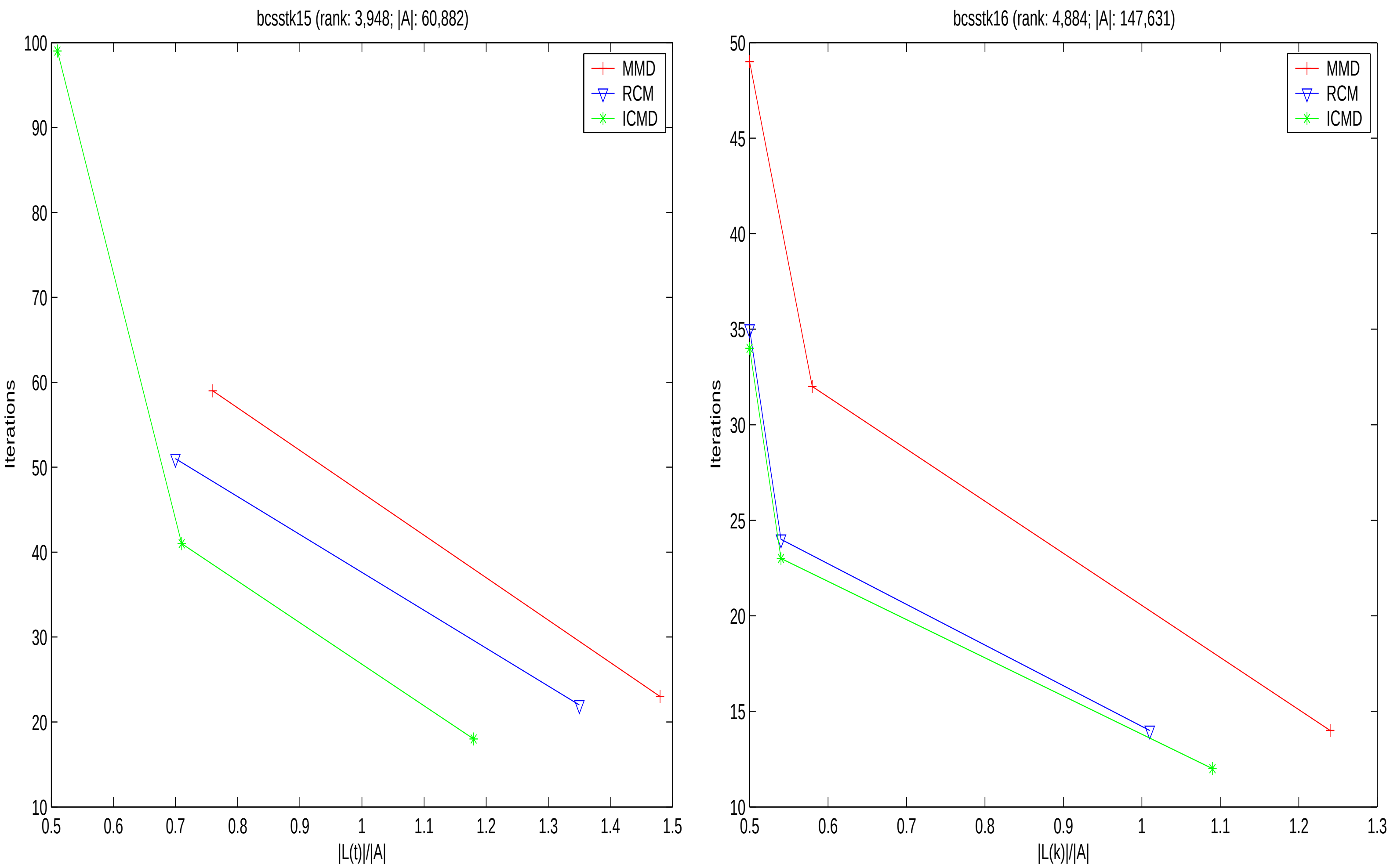


Threshold IC: bus matrices

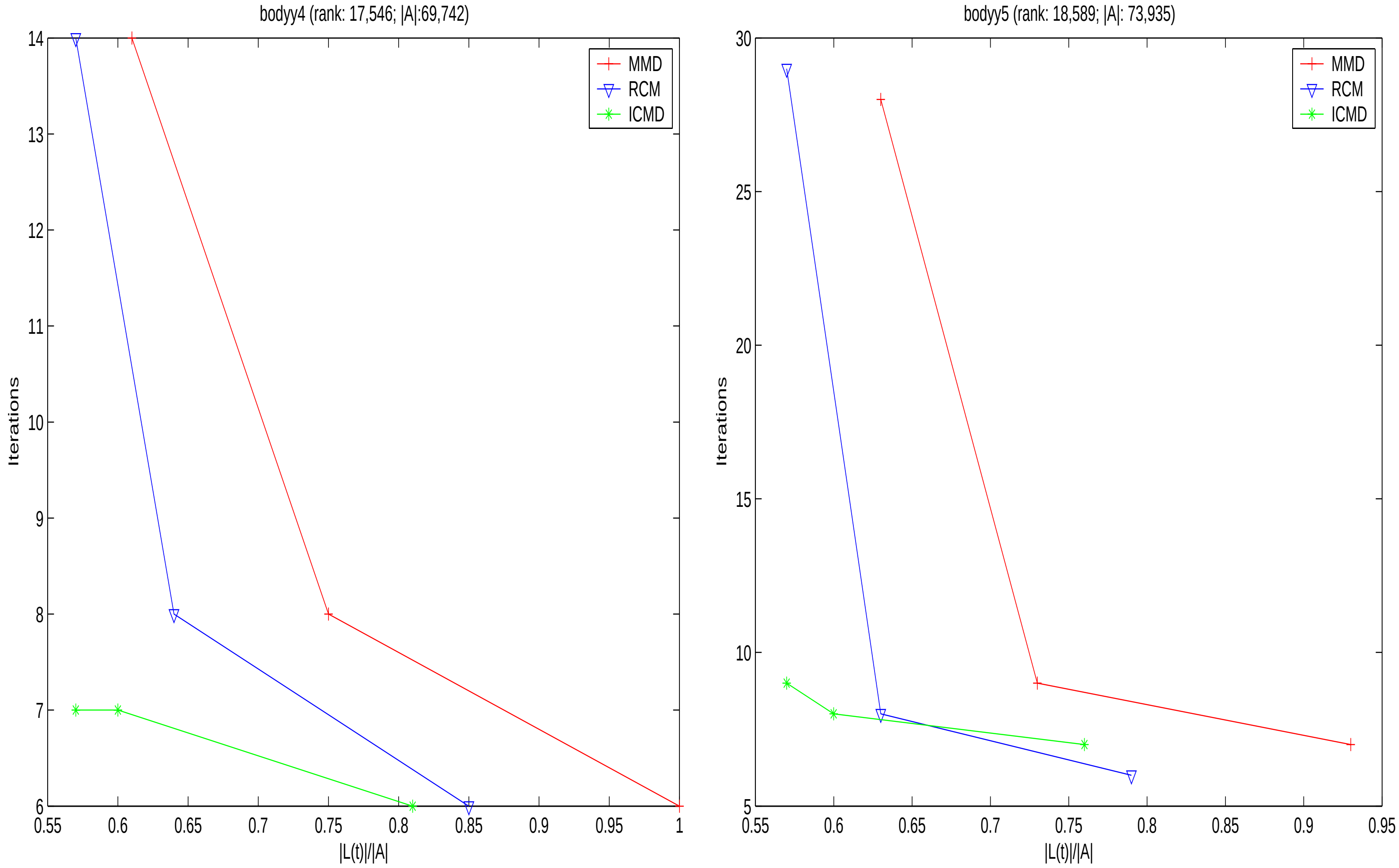




Threshold IC: `bcsstk` matrices



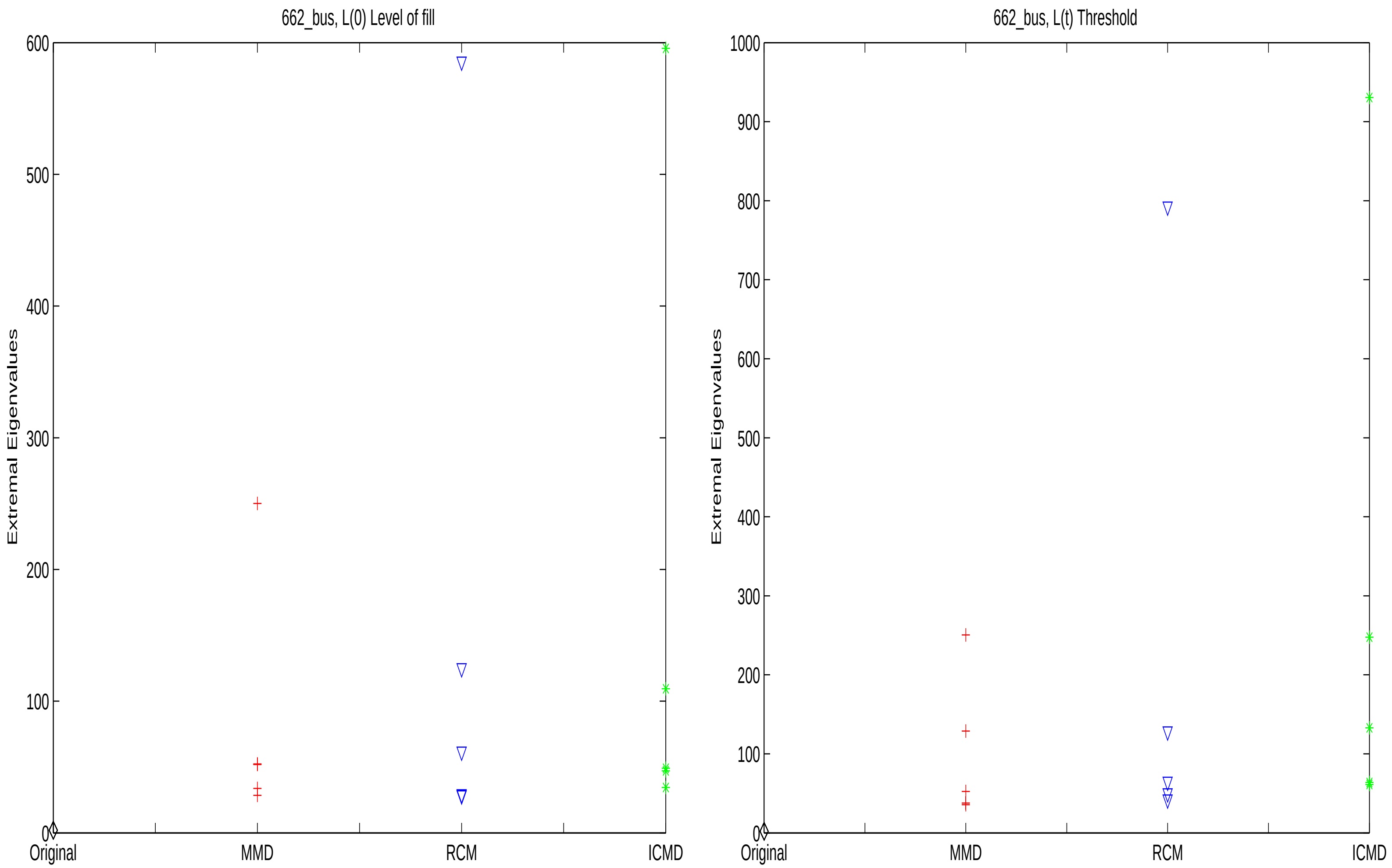
Threshold IC: bodyy matrices



A Conjecture

- Conjecture: Better convergence of PCG with ICMD $\hat{L}$ relates to its effect on the extremal eigenvalues of the preconditioned matrix $M^{-1}A$
- Pete Stewart, Numer. Math Vol 24: If λ_1 is well separated from $[\alpha, \beta]$ the interval containing $[\lambda_2, \lambda_3, \cdots \lambda_n]$, then the component of the error vector along the eigenvector corresponding to λ_1 will decrease rapidly even when the matrix is moderately ill-conditioned

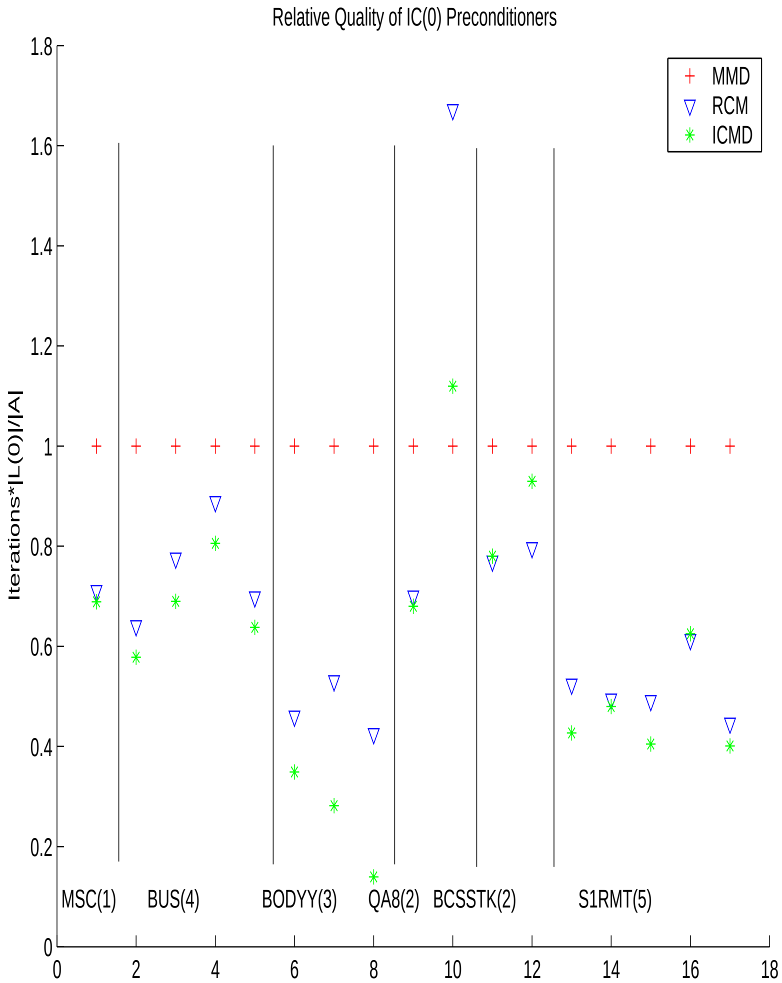
Extremal Eigenvalues

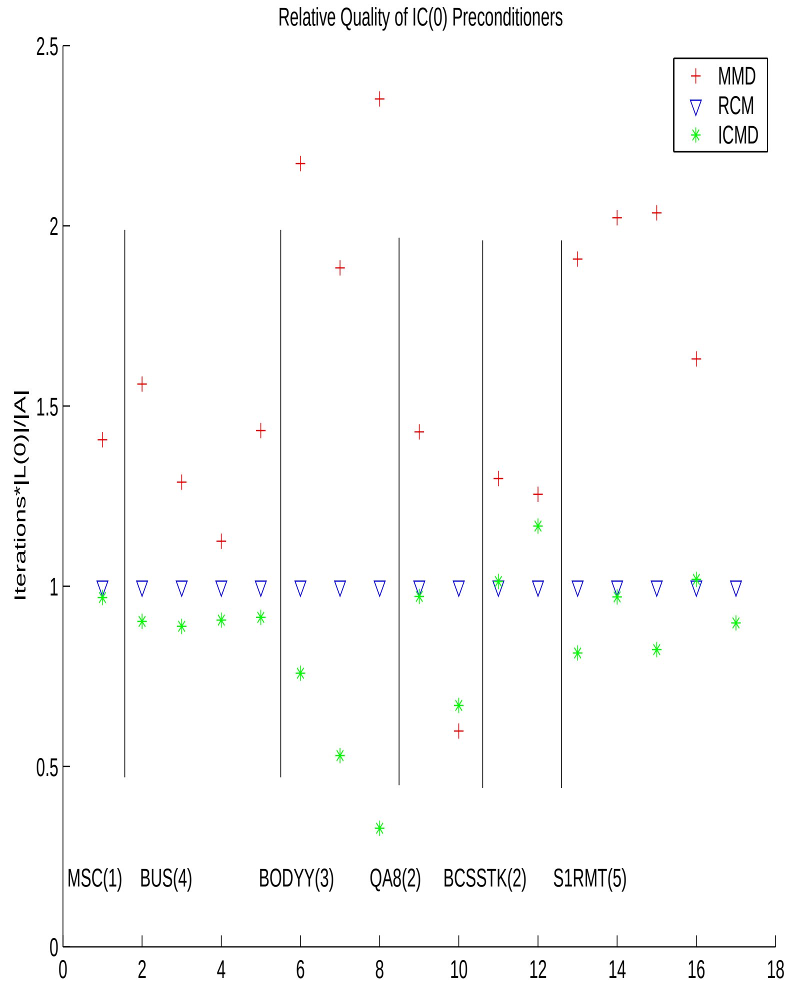


Relative Performance Summary

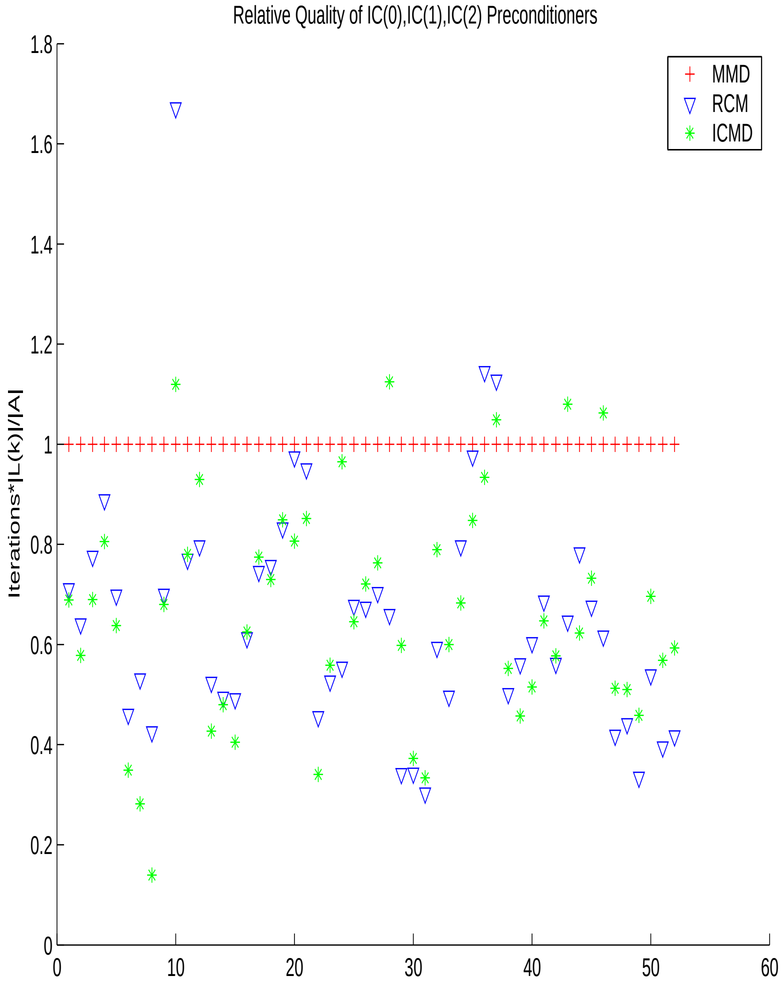
- We summarize performance for level of fill and threshold methods using scatter plots
- We define the metric μ as the product of iteration count and number of non-zeroes in $\hat{L}$
- We plot μ_{ICMD}, μ_{MMD}, μ_{RCM} along Y-axis with problem instances along X-axis
- The metric for ICMD (μ_{ICMD}) is shown normalized with respect to MMD (left) and RCM (right)

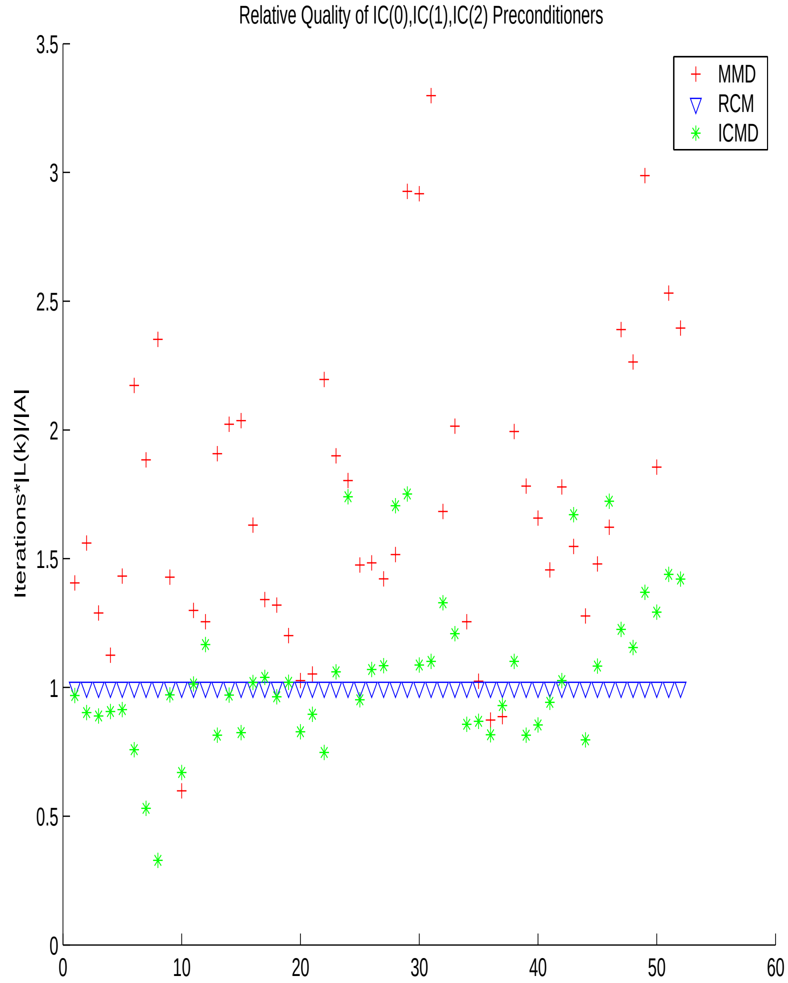
Relative Performance: Level of fill 0



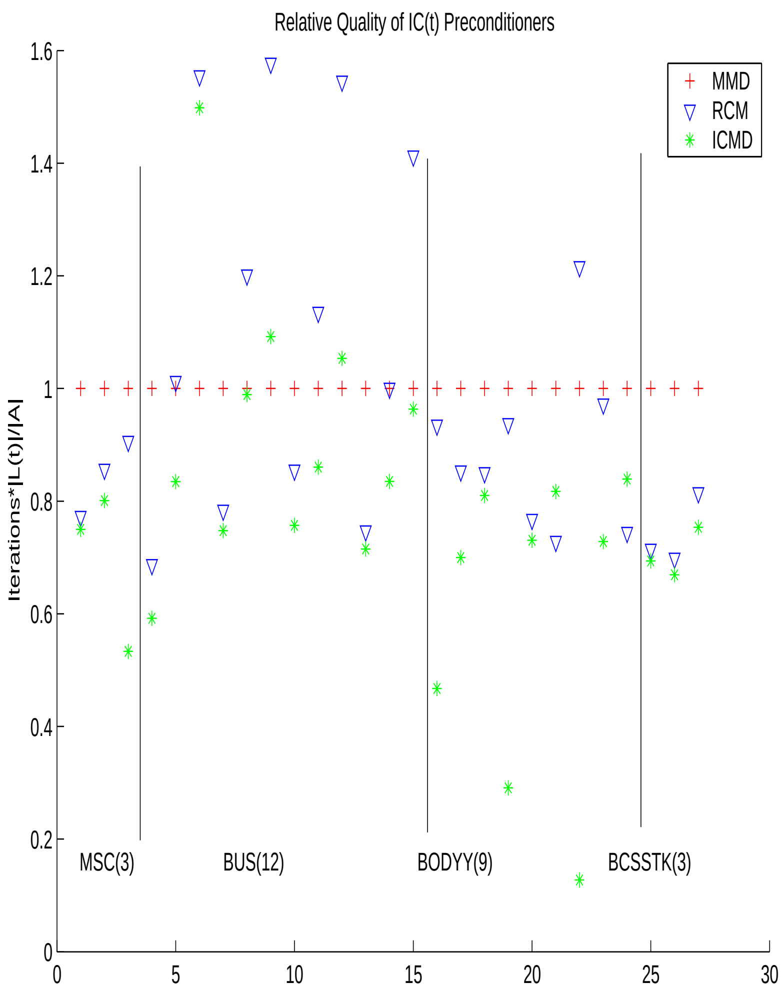


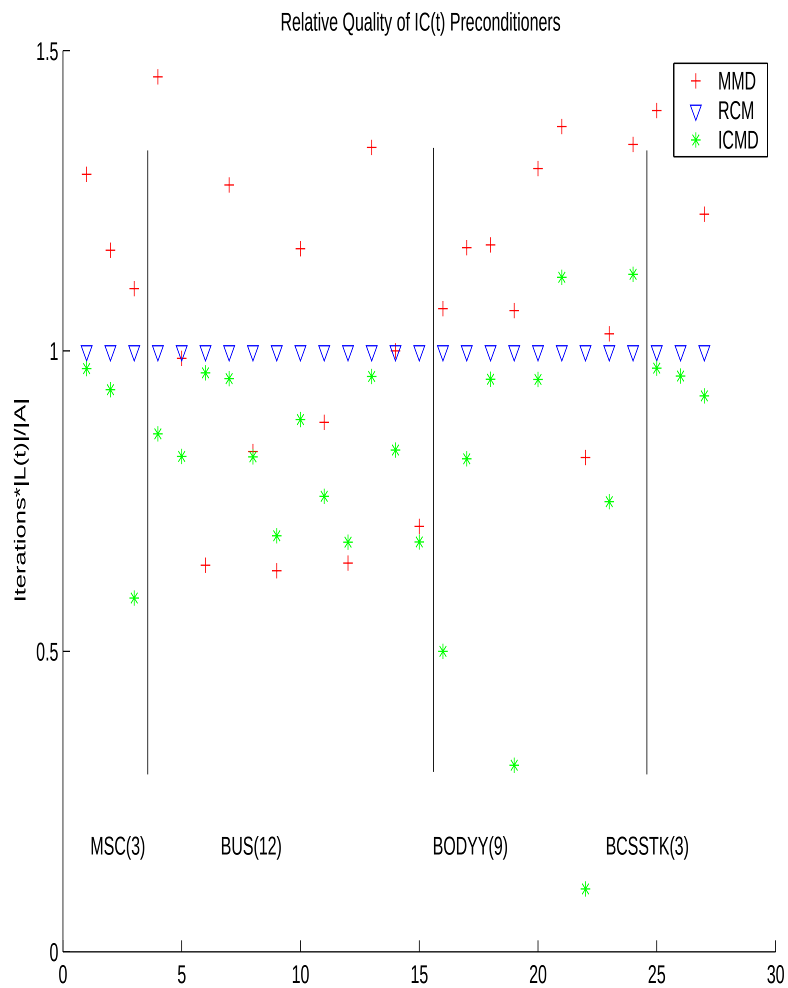
Relative Performance: Level of fill 0,1,2



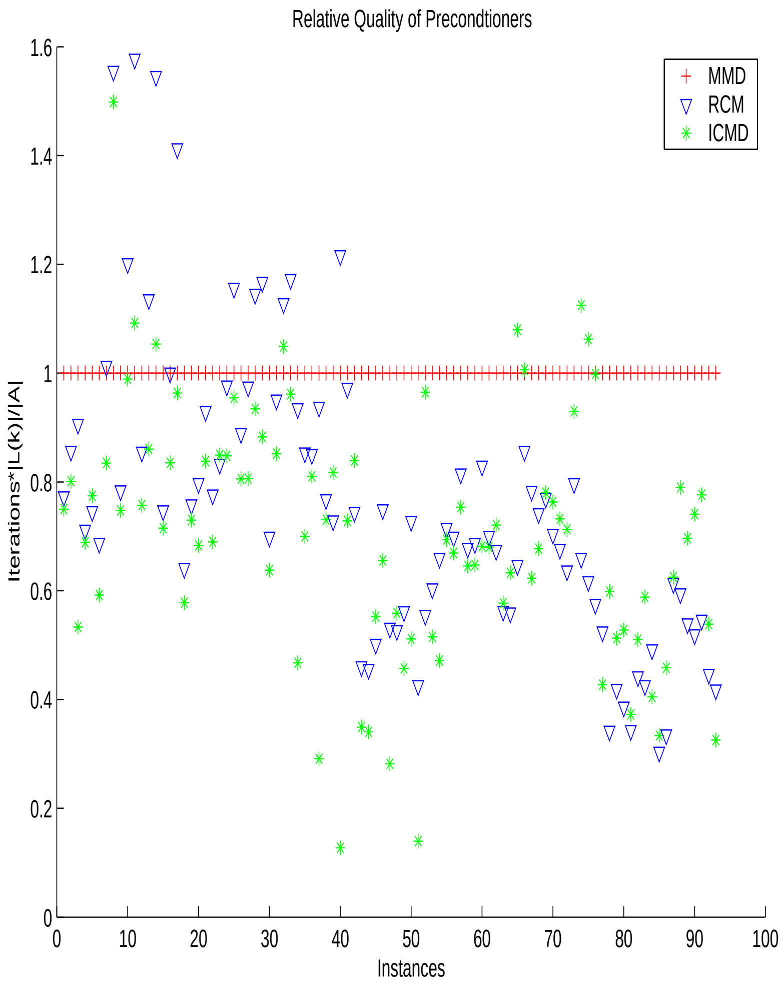


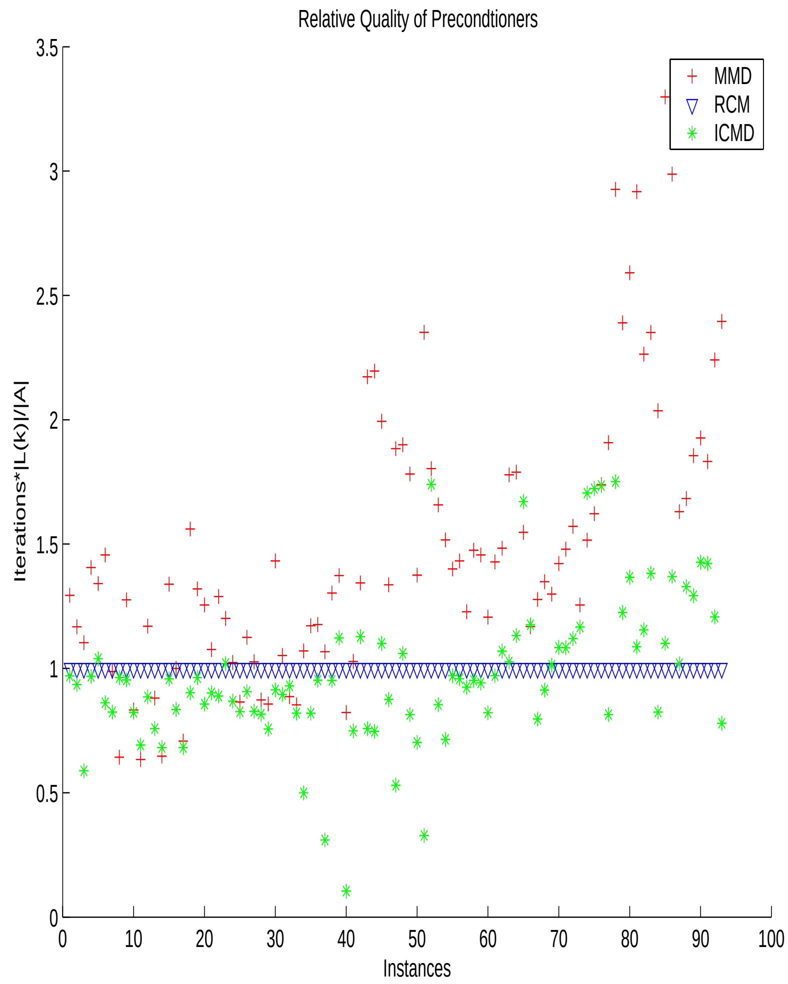
Relative Performance: 3 Thresholds



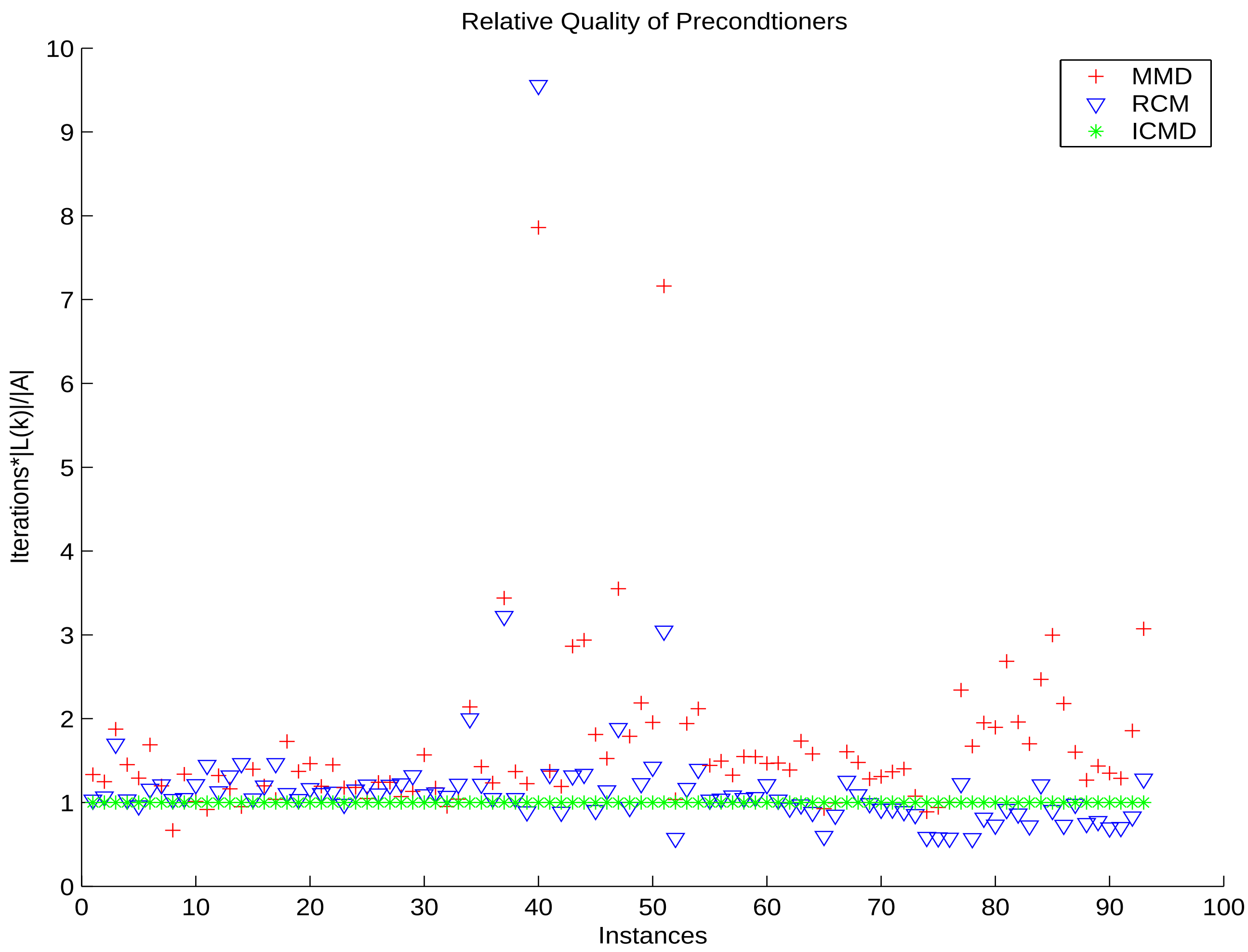


Relative Performance: All Experiments





Performance Relative to ICMD: All Experiments



Conclusions

- For **93 %** of the $\approx$ 100 instances μ_{ICMD} is lower than μ_{MMD}
- For **67 %** of the $\approx$ 100 instances μ_{ICMD} is lower than μ_{RCM}
- ICMD may not be very effective for high levels of fill $\hat{L}$ and efficient implementations will likely not be simple
- ICMD orderings seem particularly suitable for threshold schemes and for level of fill schemes with small levels
- Initial results indicate that the overheads of computing the ordering in conjunction with the numeric factorization should not be significant (for threshold and low levels of fill)