Adaptive model trust region methods for generalized eigenvalue problems

Pierre-Antoine Absil, Christopher G. Baker,

Kyle A. Gallivan

School of Computational Science, Florida State University

Ahmed Sameh

Department of Computer Sciences, Purdue University

Reminder: The Problem

Given $n \times n$ matrix pencil (A, B), $A = A^T$, $B = B^T \succ 0$ with (unknown) eigen-decomposition

$$A\,[v_1|\ldots|v_n] = B\,[v_1|\ldots|v_n]\,\mathrm{diag}(\lambda_1,\ldots,\lambda_n)$$

$$[v_1|\ldots|v_n]^T B\,[v_1|\ldots|v_n] = I, \quad \lambda_1 < \lambda_2 \le \ldots \le \lambda_n.$$

Compute the minor p-dimensional eigenspace $\mathrm{col}(v_1|\ldots|v_p)$.

The Basic Algorithm

for $k = 0, 1, 2, \ldots$

- Obtain $\mathbf{t}_k$ by approximately solve the trust-region subproblem

$$\min_{x_k^T B\mathbf{t}=0} m_{x_k}(\mathbf{t}) \quad \text{s.t. } \|\mathbf{t}\| \leq \Delta_k,$$

$$m_{x_k}(\mathbf{t}) := f(x_k) + 2\langle PAx_k, \mathbf{t}\rangle + \frac{1}{2}\langle \mathcal{H}_{x_k}[\mathbf{t}], \mathbf{t}\rangle.$$

- Update the iterate: $\rightsquigarrow x_{k+1} := (x_k + \mathbf{t}_k)/\|x_k + \mathbf{t}_k\|_B$ or $x_{k+1} := x_k$.

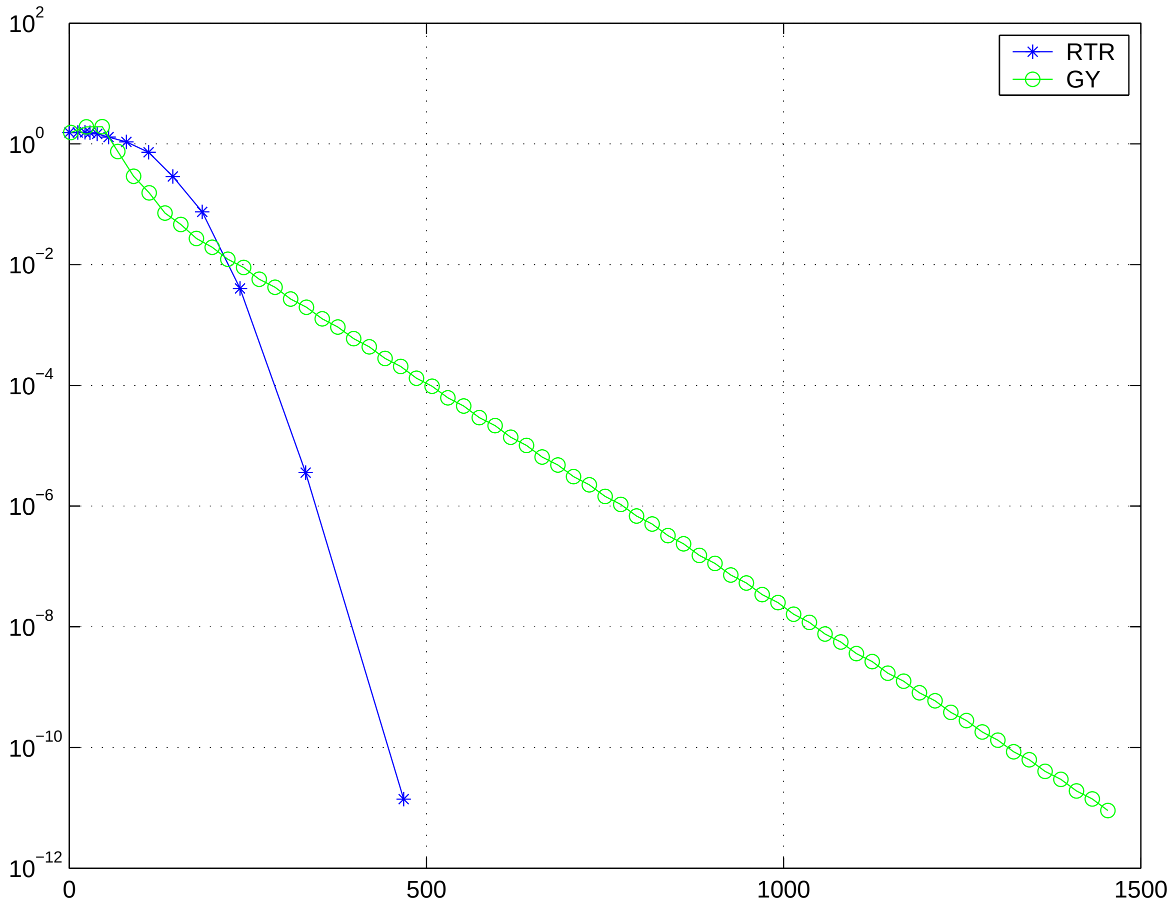
- Update the trust-region radius: $\rightsquigarrow \Delta_{k+1}$.

end (for).

Questions

- Performance of basic algorithm ?
- Enhancements ?
- Relationship to well-known methods ?

Numerical experiments: RTR vs Krylov



Distance to target versus matrix-vector multiplications.

Symmetric/positive-definite generalized eigenvalue problem.

Enhancements

To improve the performance of basic RTR:

1. Modify local model $m_x(\mathbf{t})$
2. acceleration
3. Change cost function $f(x)$

We consider the first two.

Choice of the Model

$$m_{x_k}(\mathbf{t}) := f(x_k) + 2\langle PAx_k, \mathbf{t}\rangle + \frac{1}{2}\langle \mathcal{H}_{x_k}[\mathbf{t}], \mathbf{t}\rangle.$$

Two possibilities:

- Tracemin-like (Sameh et al.):

$$\mathcal{H}^{(1)}_{x_k}[\mathbf{t}] := 2PAP\mathbf{t}.$$

- Exact quadratic expansion:

$$\mathcal{H}^{(2)}_{x_k}[\mathbf{t}] := 2P(A - f(x_k)B)P\mathbf{t}.$$

where $P = I - Bx_k(x_k^T B^2 x_k)^{-1} x_k^T B$.

Properties of TRACEMIN Model

Assume that $A \succ 0$

Tracemin-like model Hessian: $\mathcal{H}_{x_k}[\mathbf{t}] := PAP\mathbf{t}$.

- $\mathcal{H}_{x_k}[\mathbf{t}] \succ 0 \Rightarrow$ the stationary point of the model is a minimizer.
- $m_x(\mathbf{t}) \leq m_x(0_x) \Rightarrow f\left(\frac{x+\mathbf{t}}{\|x+\mathbf{t}\|_B}\right) \leq f(x)$: any decrease in the model produces a decrease in the true cost function.

Consequence: $\Delta = \infty$ – the trust-region is switched off.

Advantage: Allows, in the first outer steps, a large $\mathbf{t}_k$.

Two-phase Strategy

- Reliable but linear first phase
- Switch when in domain of attraction of second phase method
- Sameh and Wisniewski 1982 discussed this without reliable second phase
- Szyld 1988 inverse and Rayleigh quotient iterations
- Alternatives
  - characterize domains of attraction (Absil, Sepulchre, Van Dooren and Mahony 2004)
  - use reliable superlinear second phase algorithm

Two-phase Strategy

for $k = 0, 1, 2, \ldots$

- Obtain $\mathbf{t}_k$ as approximate solution of

$$\min_{x_k^T B\mathbf{t}=0} m_{x_k}(\mathbf{t}) \quad \text{s.t. } \|\mathbf{t}\| \leq \Delta_k,$$

$$m_{x_k}(\mathbf{t}) := f(x_k) + 2\langle PAx_k, \mathbf{t}\rangle + \frac{1}{2}\langle \mathcal{H}_{x_k}[\mathbf{t}], \mathbf{t}\rangle.$$

- Update the trust-region radius: $\rightsquigarrow \Delta_{k+1}$.
- Update the iterate: $\rightsquigarrow x_{k+1}$

end (for).

Phase I: $\mathcal{H}_{x_k}[\mathbf{t}] := \mathcal{H}^{(1)}_{x_k}[\mathbf{t}] = PAP\mathbf{t}$, $\Delta_0 := +\infty$.

Phase II: $\mathcal{H}_{x_k}[\mathbf{t}] := \mathcal{H}^{(2)}_{x_k}[\mathbf{t}]$ and $\Delta_k < \infty$.

Adaptive Model Example

Problem: Calgary Olympic Saddledome arena matrices from Matrix Market:

- A =BCSSTK24 and B =BCSSTM24.
- $n = 3562$
- Compute the $p = 5$ leftmost eigenpairs of (A, B)

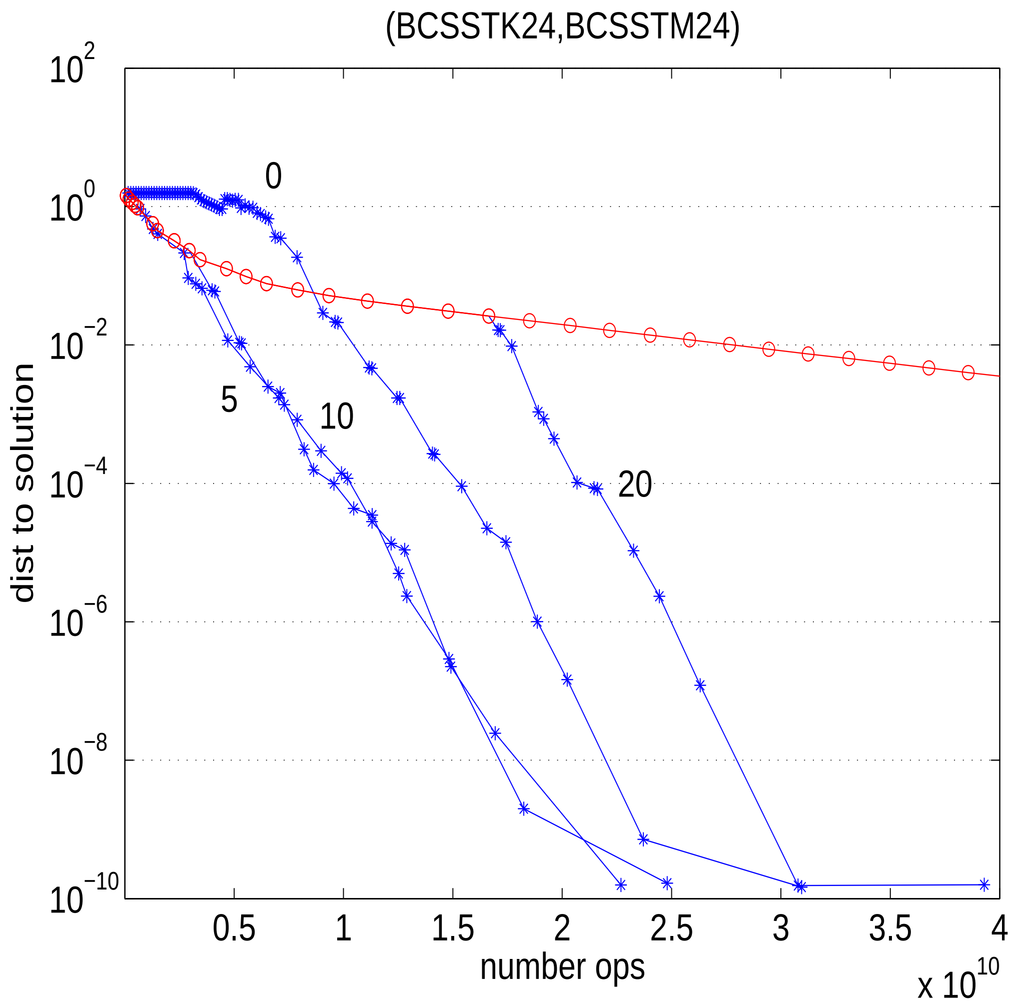


Figure 1: Angle vs. Operations – IC

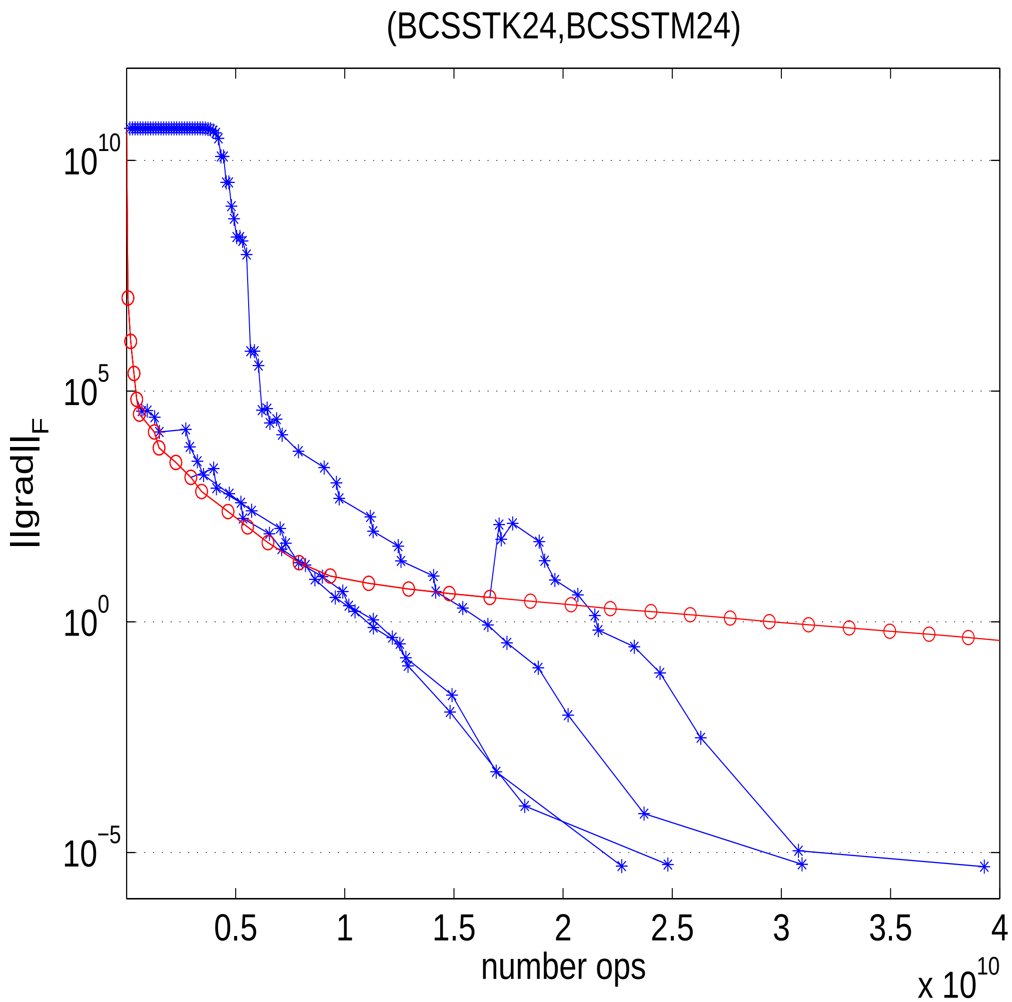


Figure 2: Gradient vs. Operations – IC

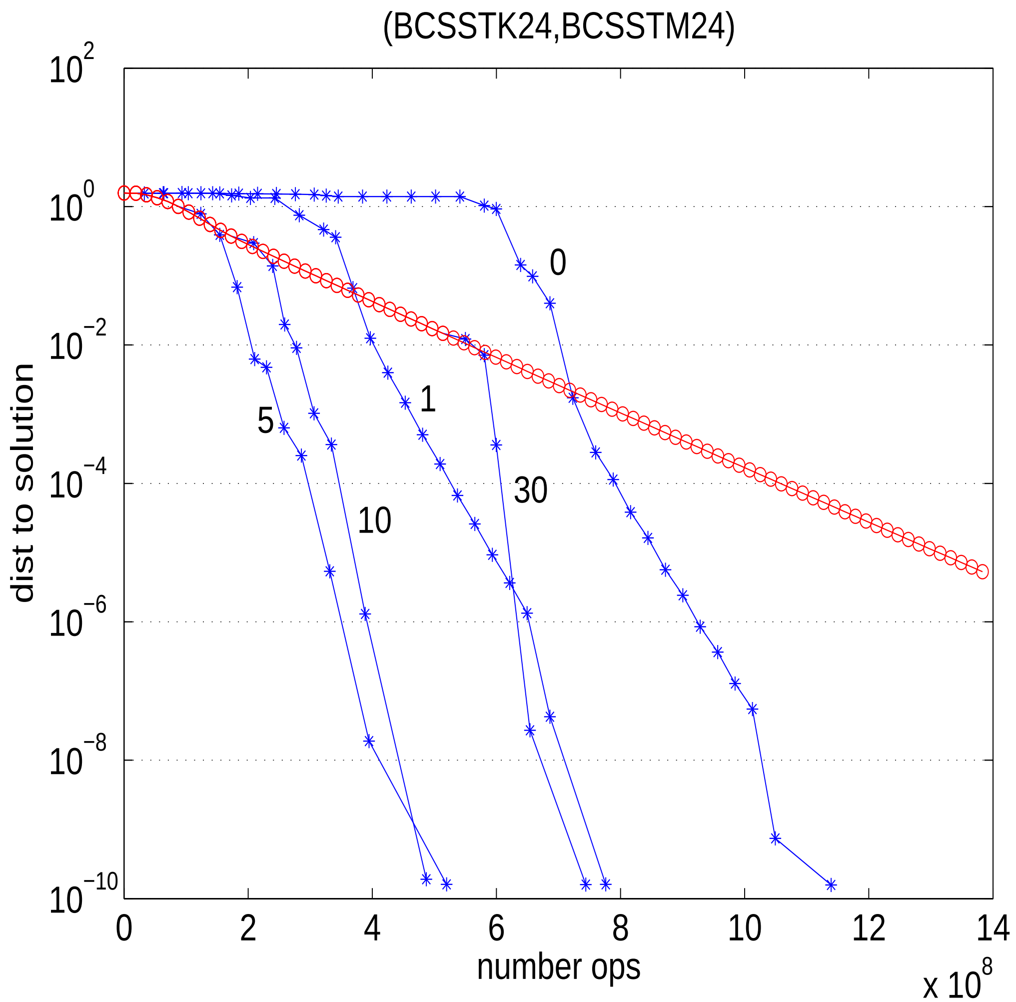


Figure 3: Angle vs. Operations – AMD, Chol

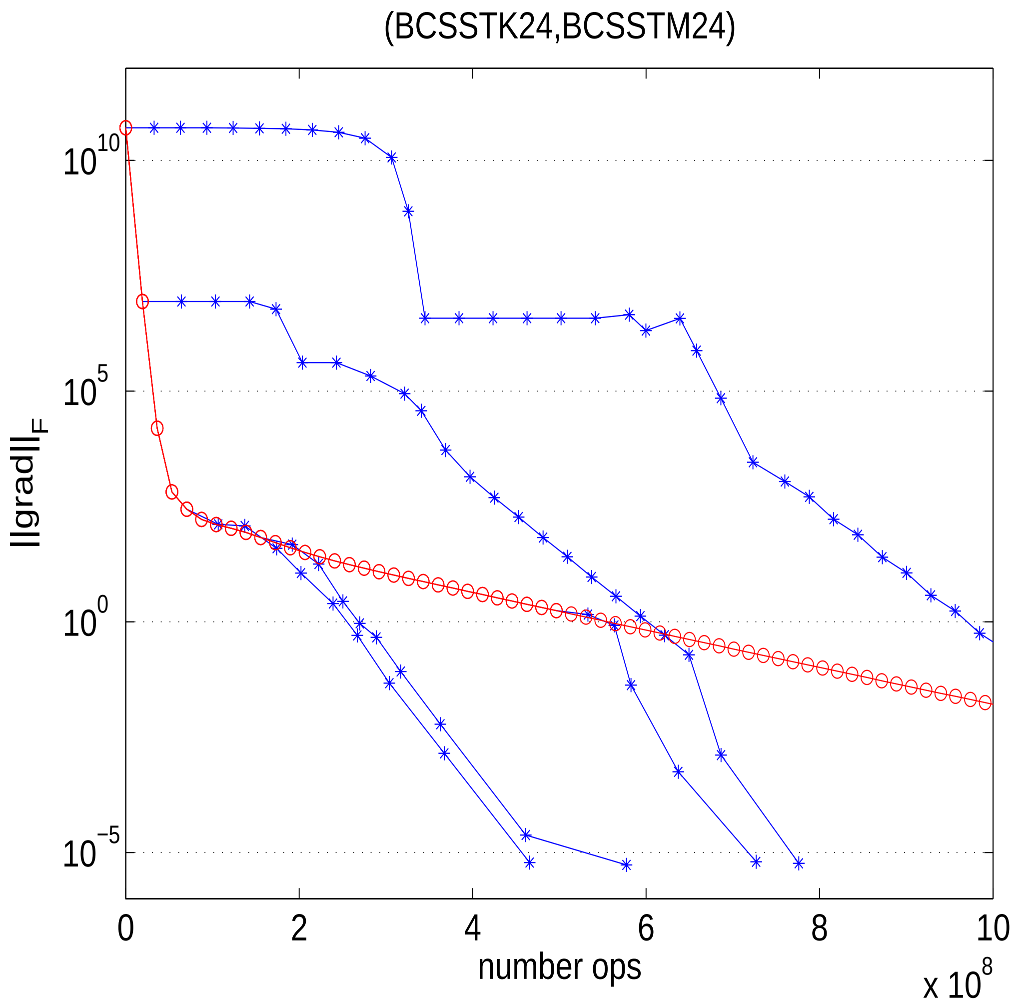


Figure 4: Gradient vs. Operations – AMD, Chol

Adaptive Model

- efficiency not convergence
- rapid drop in f and its gradient
- switch when leveling off seen
- switch back (?)
- adaptive model for indefinite problems (?)

Second enhancement: subspace acceleration

- acceleration included in manifold theory
- sufficient condition on spaces and iteration for global convergence
- consistent with sequential and expand-restart subspace acceleration
- works with RTR
- investigating applicability to other methods

Second enhancement: subspace acceleration

for $k = 0, 1, 2, \ldots$

- Obtain $\mathbf{t}_k$ as approximate solution of

$$\min_{x_k^T B\mathbf{t}=0} m_{x_k}(\mathbf{t}) \quad \text{s.t. } \|\mathbf{t}\| \leq \Delta_k,$$

- Update the trust-region radius: $\rightsquigarrow \Delta_{k+1}$.
- Pick subspace $\mathcal{V}_k$ based on $\mathbf{t}_k$ and previous information.
- Choose $x_{k+1} = \arg\min_{x \in \mathcal{V}_k \cap \mathcal{M}} f(x)$.

end (for).

Second enhancement: SA: "Sequential Subspace" approach

Input: Acceleration subspace basis $V_k = [X_k, \mathbf{t}_k, P_{k-b+3}, \ldots, P_k]$ of dimension $m = b * p$.

Output: iterate X_{k+1}; new search direction P_{k+1}

Ritz-Rayleigh procedure

Form $\hat{A} = V_k^T A V_k$ and $\hat{B} = V_k^T B V_k$

Compute p smallest eigenpairs $(u_k, \hat{\lambda}_k)$ of $(\hat{A}, \hat{B})$

$U_k := [u_1| \ldots |u_p]$

Compute next iterate

$X_{k+1} = V_k U_k = X_k \hat{U}_k + [\mathbf{t}_k, P_{k-b+3}, \ldots, P_k]\tilde{U}_k$

Compute next search direction

$P_{k+1} = [\mathbf{t}_k, P_{k-b+3}, \ldots, P_k]\tilde{U}_k$

Second enhancement: SA: "Expand-Restart" approach

Input: Acceleration subspace basis V_k; update $\mathbf{t}_k$.

Output: iterate X_{k+1}; new subspace basis V_{k+1}.

Orthonormalization

$W := \mathrm{orth}\left([V_k|\mathbf{t}_k]\right)$

Ritz-Rayleigh procedure

Form $\hat{A} = W^T A W$ and $\hat{B} = W^T B W$

Compute p smallest eigenpairs $(u_k, \hat{\lambda}_k)$ of $(\hat{A}, \hat{B})$

$U_k := [u_1|\dots|u_p]$

$X_{k+1} := V_k U_k$

Restart

If $\dim(W) \geq m_{\max}$

$V_{k+1} = W$

else

$V_{k+1} = X_{k+1}$

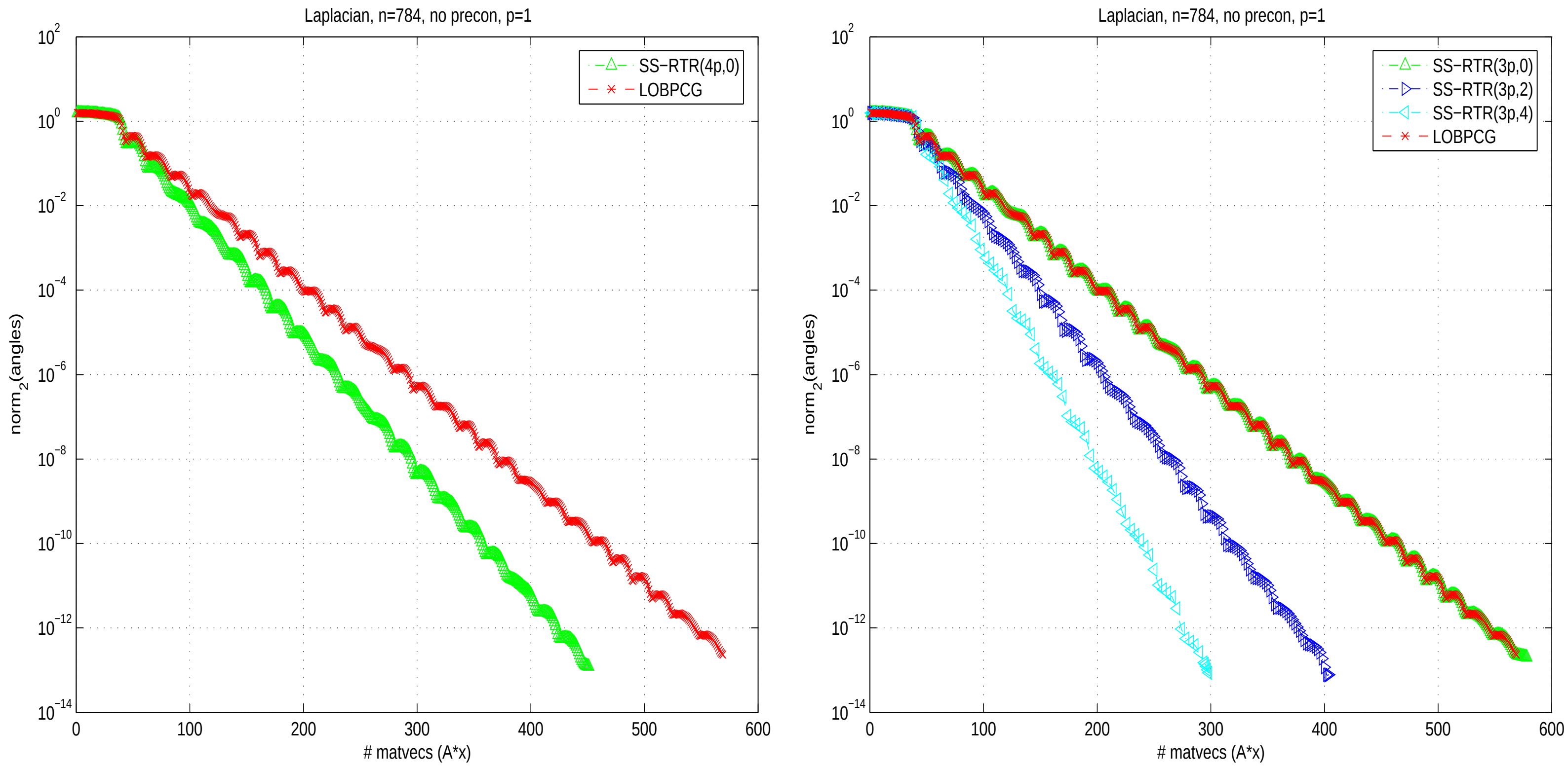


Figure 5: Acceleration Angle vs. Mxv – Laplacian

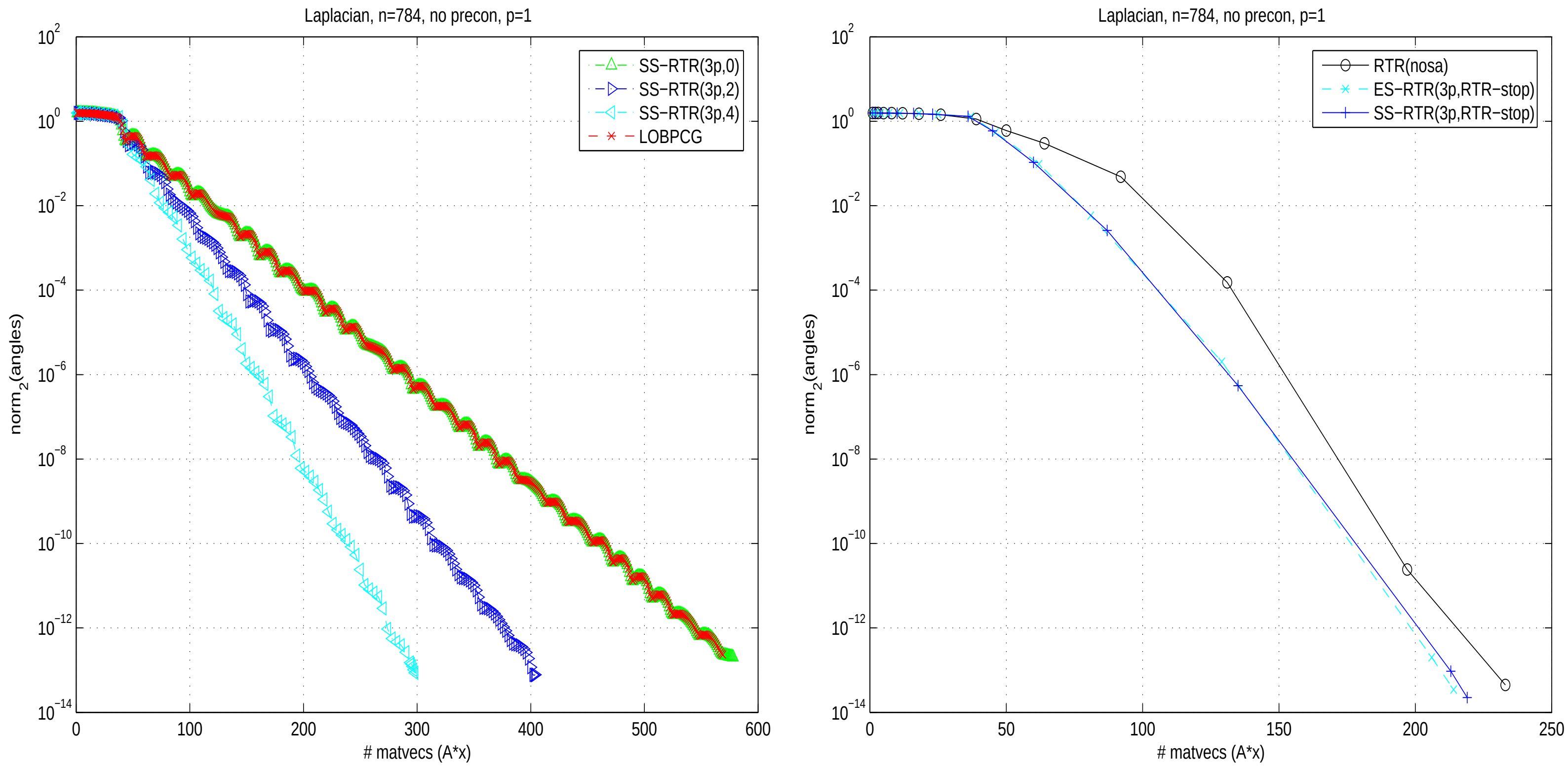


Figure 6: Acceleration Angle vs. Mxv – Laplacian

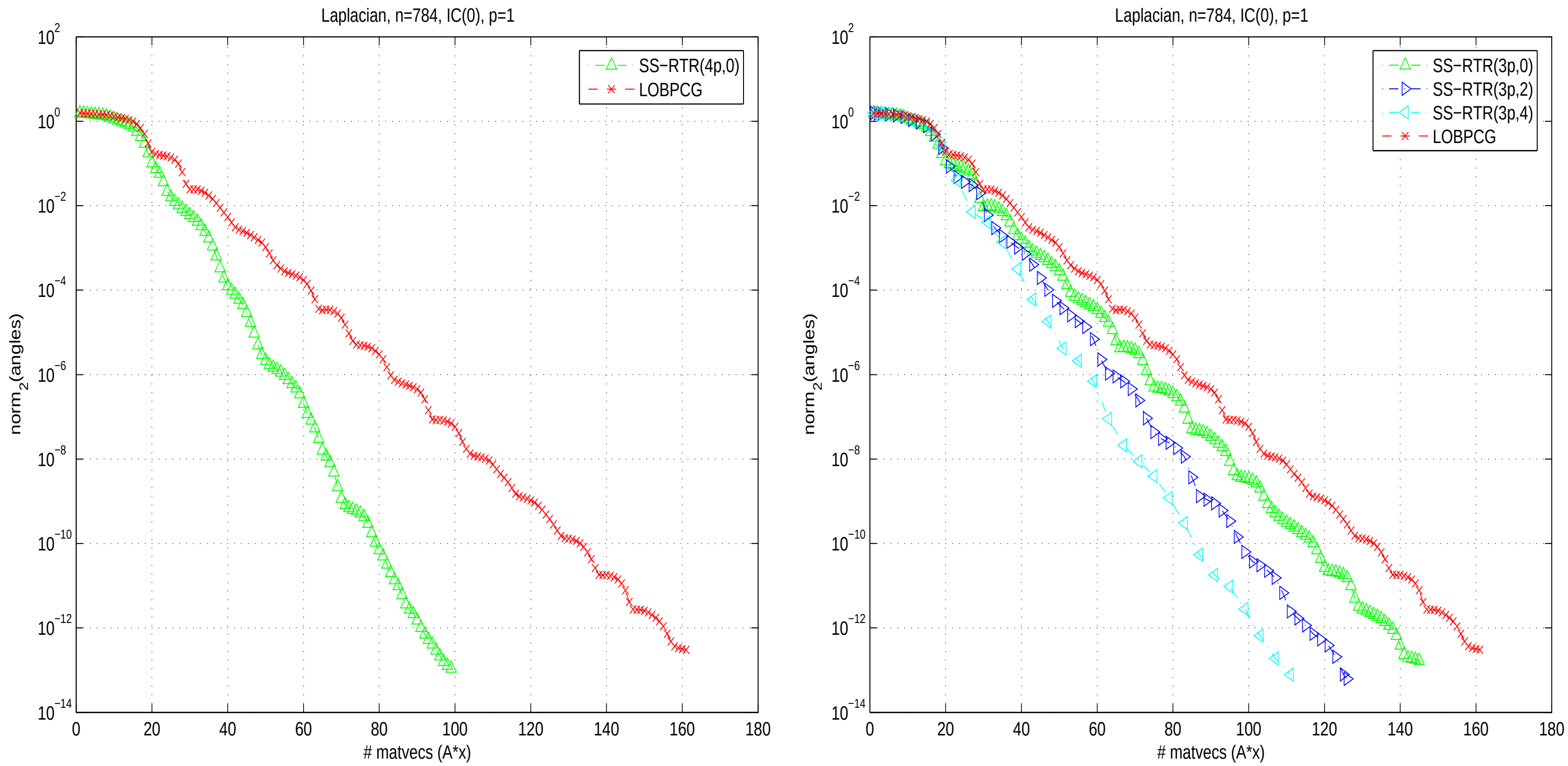


Figure 7: Acceleration Angle vs. Mxv – Laplacian IC(0)

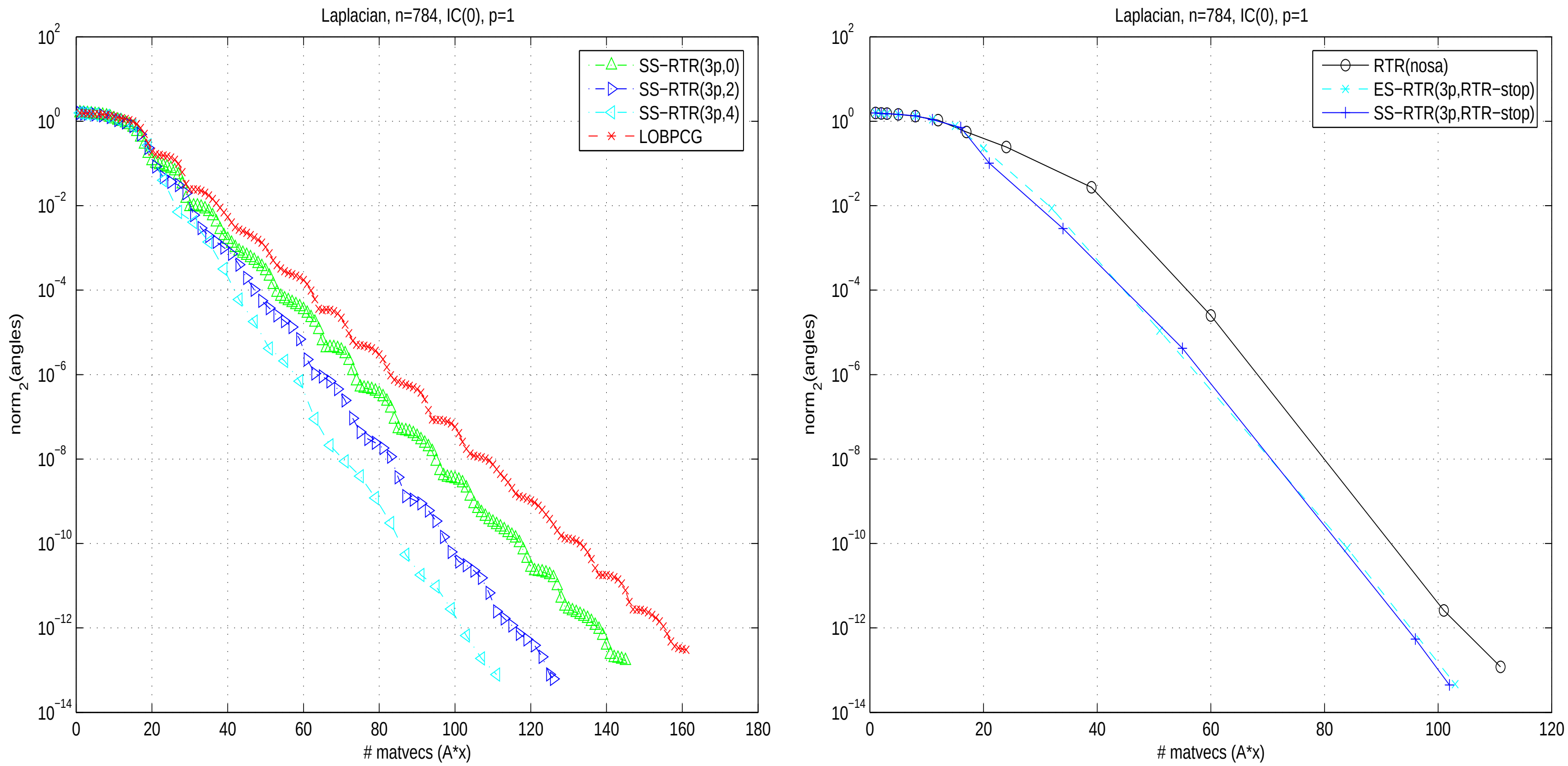


Figure 8: Acceleration Angle vs. Mxv – Laplacian IC(0)

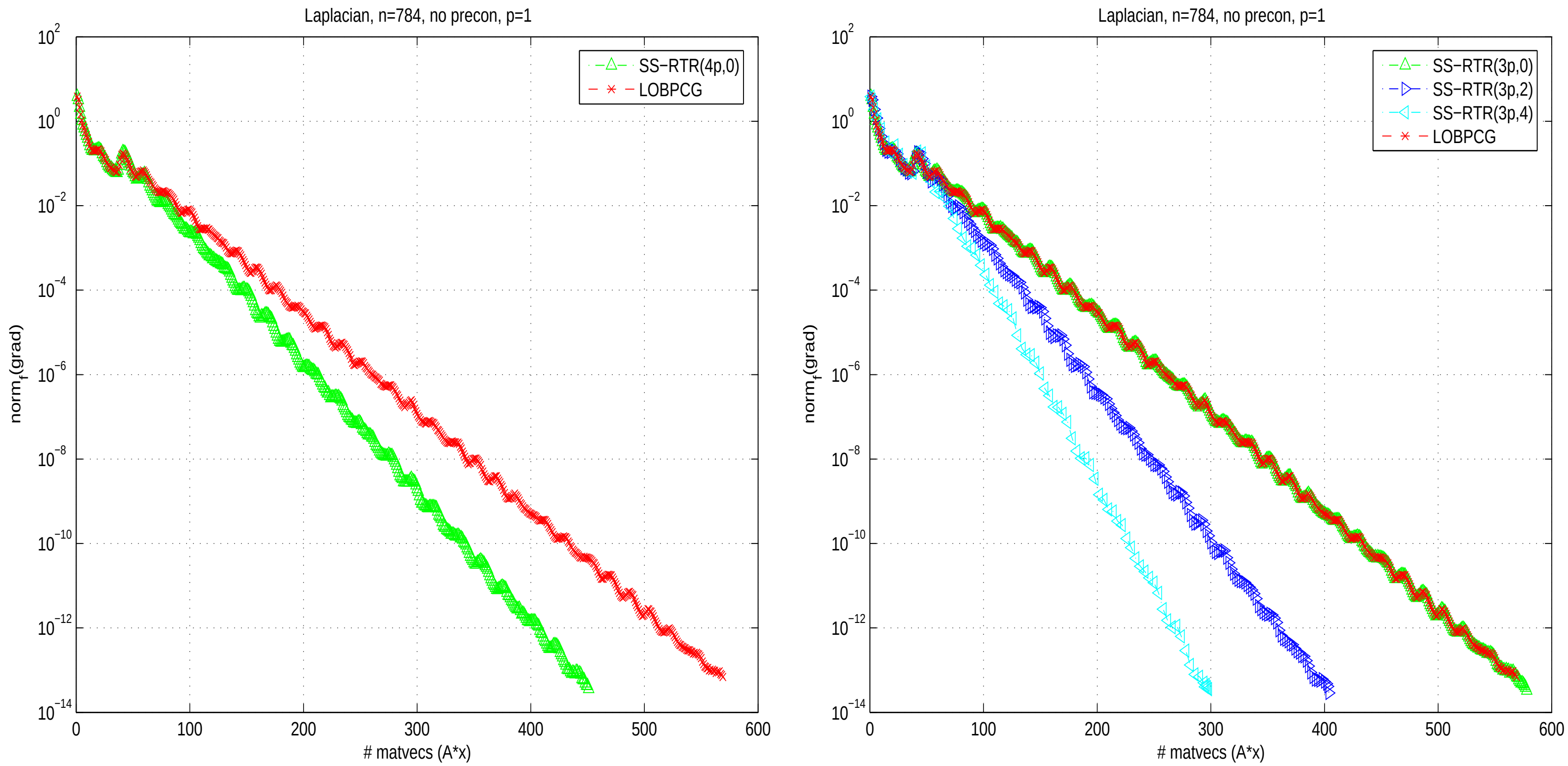
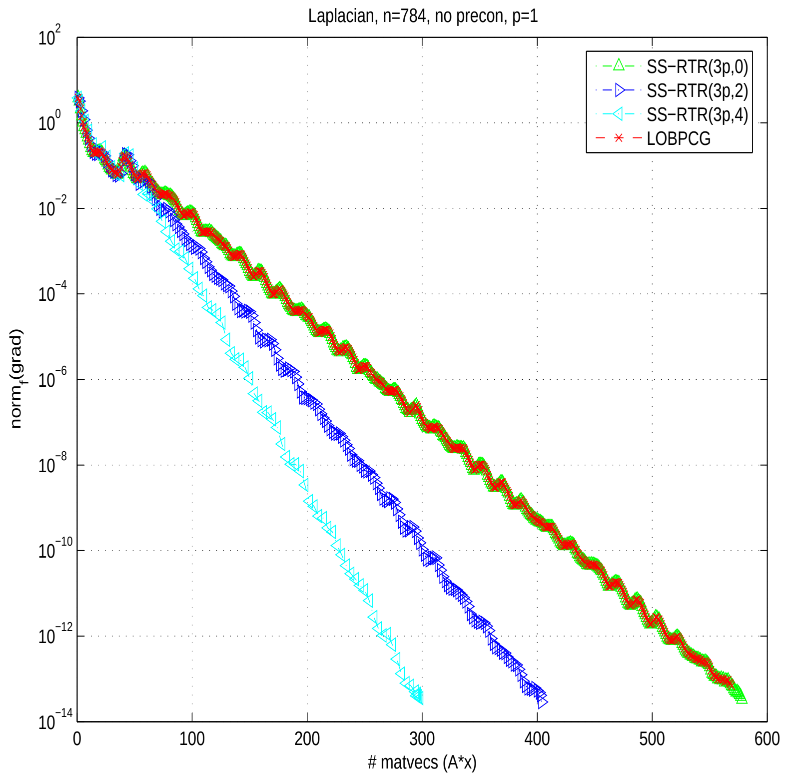


Figure 9: Acceleration Gradient vs. Mxv – Laplacian



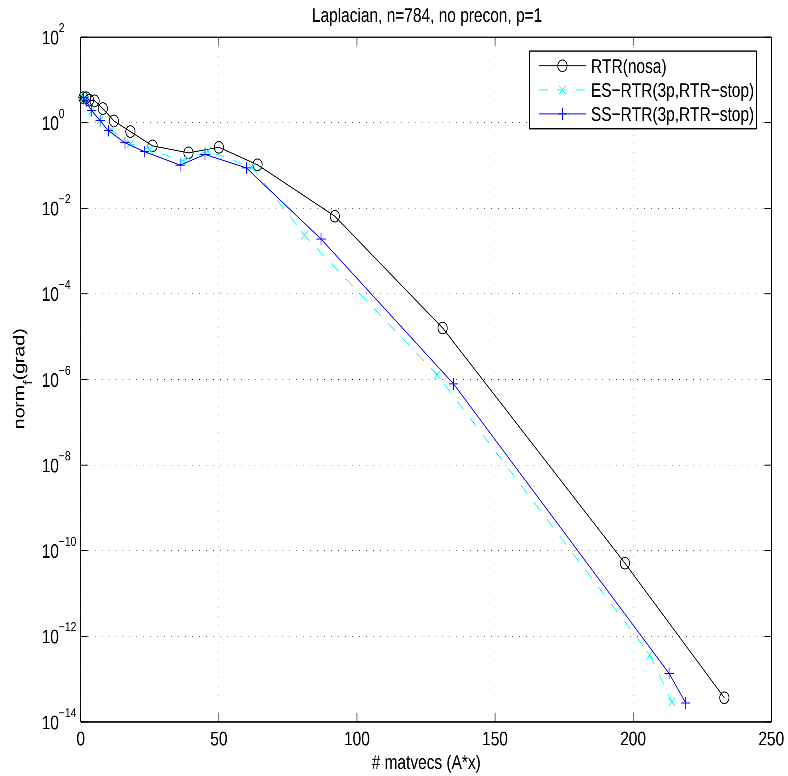
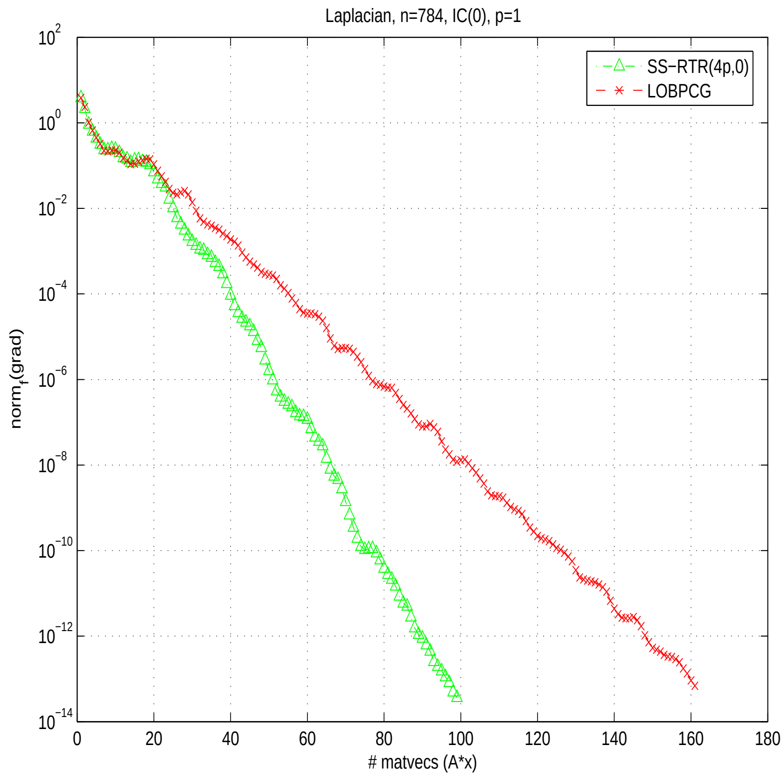


Figure 10: Acceleration Gradient vs. Mxv – Laplacian



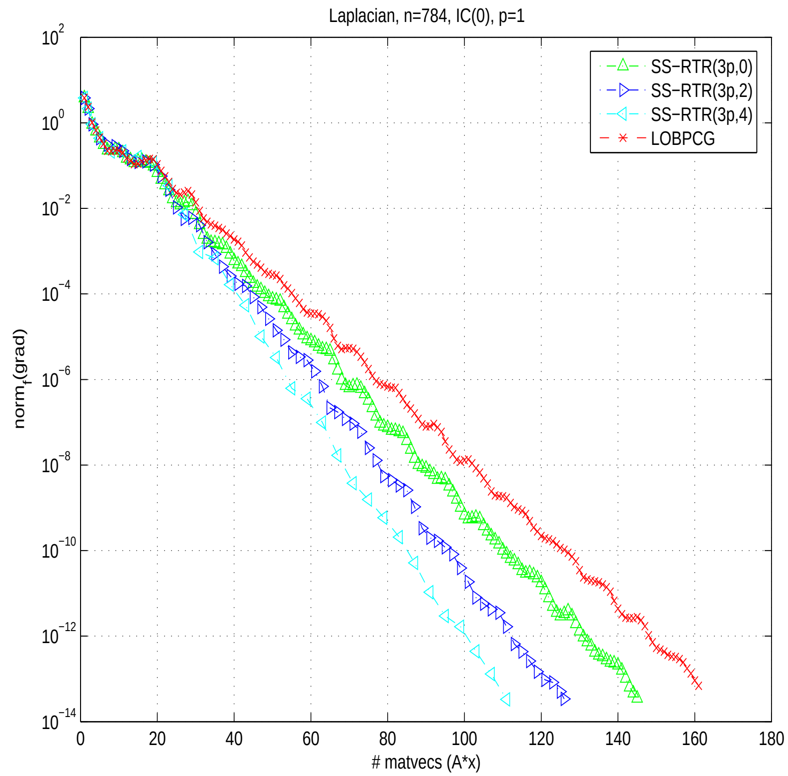


Figure 11: Acceleration Gradient vs. Mxv – Laplacian IC(0)

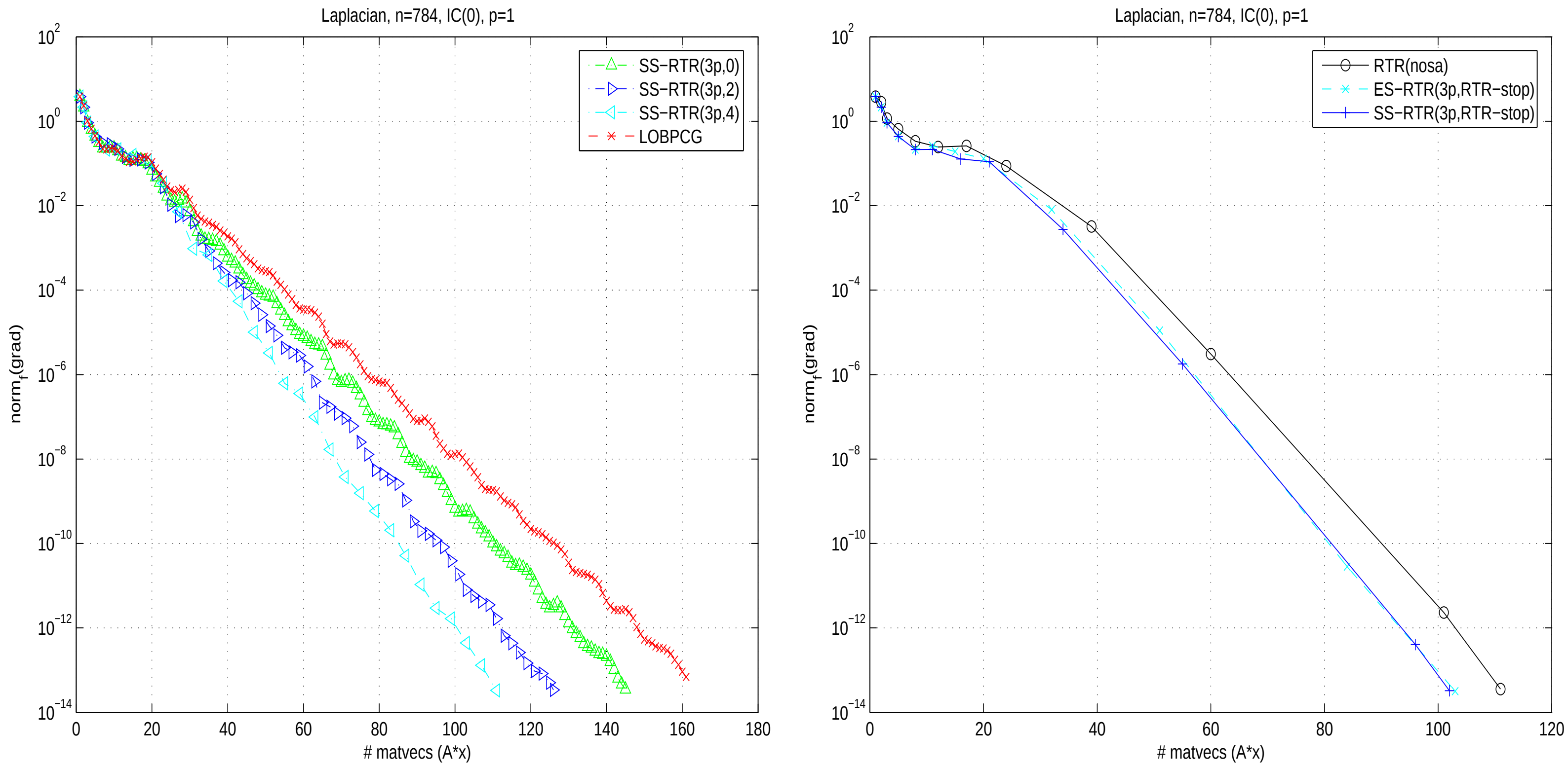


Figure 12: Acceleration Gradient vs. Mxv – Laplacian IC(0)

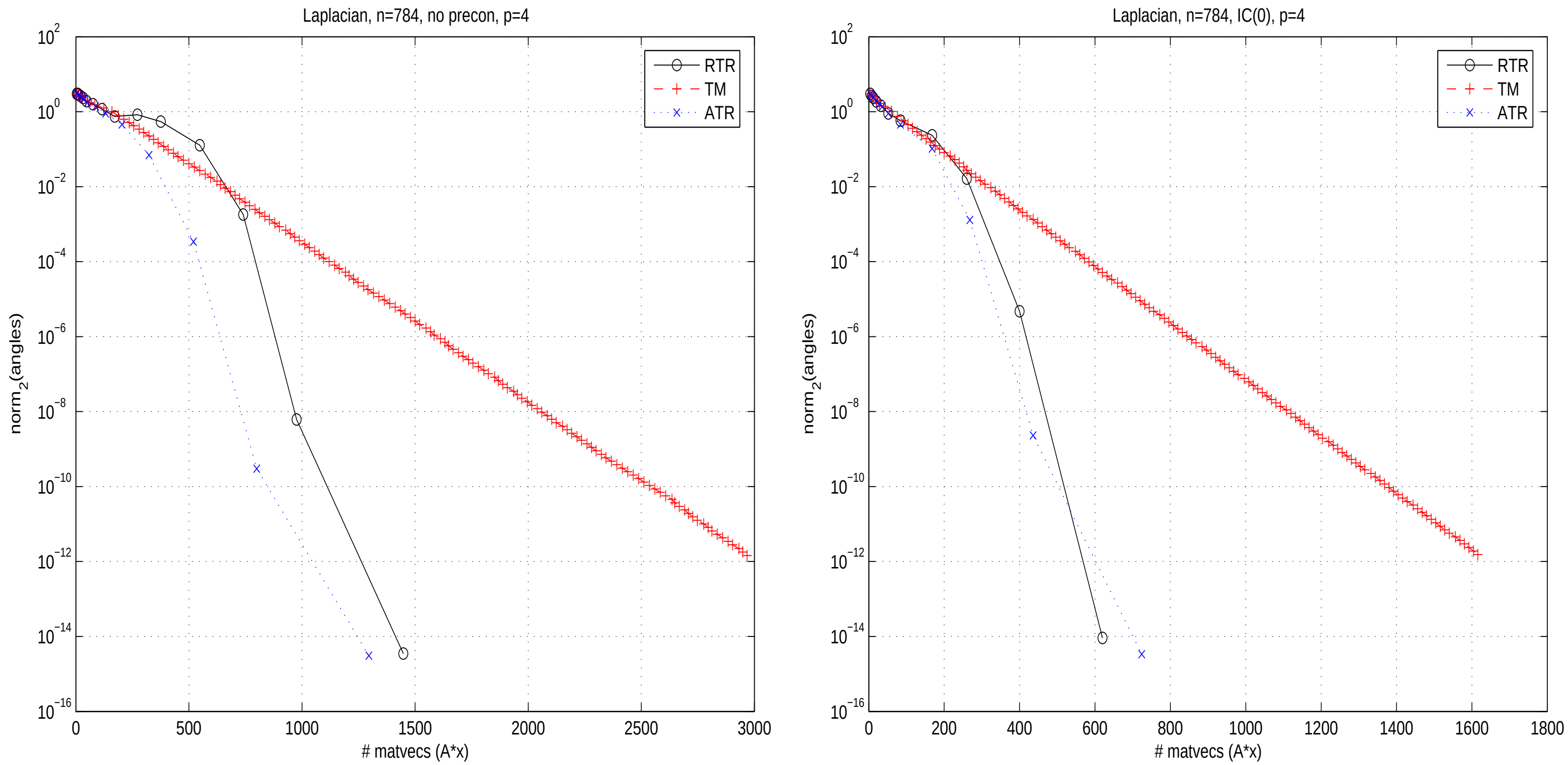


Figure 13: Adaptive Model Angle vs. Mxv – Laplacian

Conclusions and Future Work

- Adaptive model promising for positive definite problems (indefinite?)
- Switch evaluation
- Subspace acceleration can be useful
- Careful implementation (including semidefinite)
- Systematic evaluation and comparison
- Subspace acceleration and non-TR iterations
- Adaptive model + subspace acceleration