A Parallel Implementation of the Jacobi-Davidson Eigensolver for Unsymmetric Matrices

Eloy Romero[1] Manuel B. Cruz[2] Jose E. Roman[1]
Paulo B. Vasconcelos[3]

[1] I3M/UPV Spain [2] LEMA/ISEP/IPP Portugal [3] CMUP/FEP.UP Portugal

VECPAR'10 | 22 Jun/26 Jun | Berkeley, U.S.A.

Structure

A Parallel Implementation of the Jacobi-Davidson Eigensolver for Unsymmetric Matrices

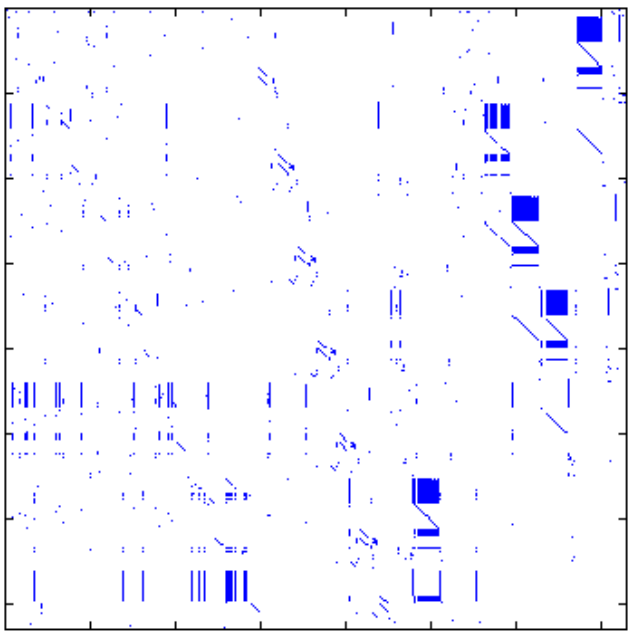
1. Introduction
2. Jacobi-Davidson Method
3. Implementation Details
4. Numerical Results
5. Conclusions and Future Work

The Eigenvalue Problem

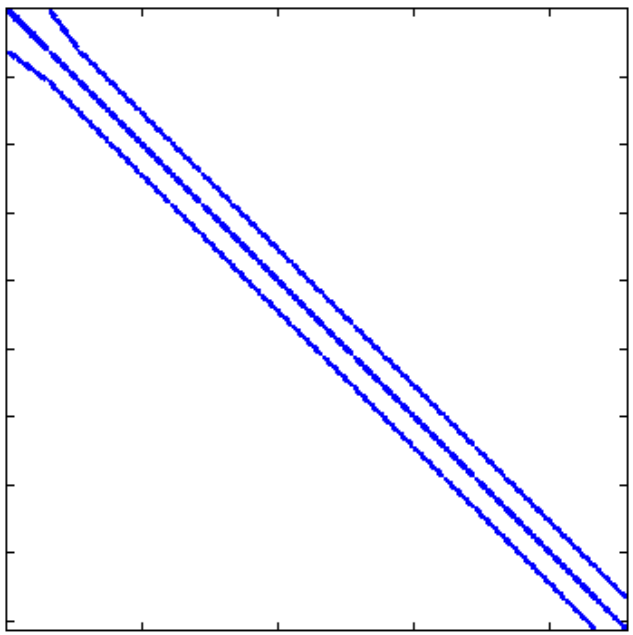
"...there are still many open problems in this exciting field and these problems involve interesting work to be done by theorists as well as more practical oriented computational scientists. [...]still needs attention when it comes to parallel implementation." *Vorst and Golub* in "150 Years old and still alive: eigenproblems"

$$Ax = \lambda x, \quad A \in \mathbb{C}^{n \times n}, \; x \in \mathbb{C}^{n \times 1}, \; \lambda \in \mathbb{C}$$

Non-symmetric Matrices.



Large Matrices.



History

1846 - Jacobi: Search for the orthogonal correction to eigenvector approximation (JOCC).

1975 - Davidson: Search subspace expansion.

1996 - Sleijpen and Vorst: Original article for the Jacobi-Davidson method (1996 SIAM Best paper in Numerical Linear Algebra).

From 1996: Several variants by different authors.

There are no freely available parallel implementations for the non-hermitian case.

General Algorithm

Algorithm 1 Jacobi-Davidson

Procedure JD(A, u_0, tol, $itmax$)

$u \leftarrow u_0/\|u_0\|_2$, $V \leftarrow [u]$, $\theta \leftarrow u^* Au$, $r \leftarrow Au - \theta u$

$m \leftarrow 1$

while $\|r\|_2/|\theta| > tol$ & $m < itmax$

Solve approximately $(I - uu^*)(A - \theta I)(I - uu^*)t = -r, \quad t \perp u$

$V \leftarrow$ Orthonormalize$[V, t]$, and $m \leftarrow m + 1$

Compute a desired eigenpair (θ, s) from the projected eigensystem using standard or harmonic techniques

$u \leftarrow Vs$, $r \leftarrow Au - \theta u$

$it = it + 1$

return θ, u

- Projection methods: subspace expansion + extraction phase.

Subspace Expansion

- Ritz Pair: $(\theta, u) \Rightarrow$ Residual: $r = (A - \theta I)u \Rightarrow$ New vector t:

Correction Equation (C.E.):

$$(I - uu^*)(A - \theta I)(I - uu^*)t = -r, \quad \text{with } t \perp u$$

- GMRES with low tolerance and $t \perp u$ per iteration.
- $(I - uu^*)(A - \theta I)t = -r$, for the exterior case.
- Interior with preconditioner. C.E. solved via:

Interior Eigenvalues:

$$\widetilde{K}^{-1}\widetilde{A_\theta}t = -\widetilde{K}^{-1}r$$

where $\widetilde{A_\theta} = (I - uu^*)(A - \theta I)(I - uu^*)$ and $\widetilde{K} \approx \widetilde{A_\tau}$

Subspace Expansion

- Ritz Pair: $(\theta, u) \Rightarrow$ Residual: $r = (A - \theta I)u \Rightarrow$ New vector t:

Correction Equation (C.E.):

$$(I - uu^*)(A - \theta I)(I - uu^*)t = -r, \quad \text{with } t \perp u$$

- GMRES with low tolerance and $t \perp u$ per iteration.
- $(I - uu^*)(A - \theta I)t = -r$, for the exterior case.
- Interior with preconditioner. C.E. solved via:

Interior Eigenvalues:

$$\widetilde{K}^{-1}\widetilde{A_\theta}t = -\widetilde{K}^{-1}r$$

where $\widetilde{A_\theta} = (I - uu^*)(A - \theta I)(I - uu^*)$ and $\widetilde{K} \approx \widetilde{A_\tau}$

Subspace Expansion

- Ritz Pair: $(\theta, u) \Rightarrow$ Residual: $r = (A - \theta I)u \Rightarrow$ New vector t:

Correction Equation (C.E.):

$$(I - uu^*)(A - \theta I)(I - uu^*)t = -r, \quad \text{with } t \perp u$$

- GMRES with low tolerance and $t \perp u$ per iteration.
- $(I - uu^*)(A - \theta I)t = -r$, for the exterior case.
- Interior with preconditioner. C.E. solved via:

Interior Eigenvalues:

$$\widetilde{K}^{-1}\widetilde{A_\theta}t = -\widetilde{K}^{-1}r$$

where $\widetilde{A_\theta} = (I - uu^*)(A - \theta I)(I - uu^*)$ and $\widetilde{K} \approx \widetilde{A_\tau}$

Subspace Expansion

- Ritz Pair: $(\theta, u) \Rightarrow$ Residual: $r = (A - \theta I)u \Rightarrow$ New vector t:

Correction Equation (C.E.):

$$(I - uu^*)(A - \theta I)(I - uu^*)t = -r, \quad \text{with } t \perp u$$

- GMRES with low tolerance and $t \perp u$ per iteration.
- $(I - uu^*)(A - \theta I)t = -r$, for the exterior case.
- Interior with preconditioner. C.E. solved via:

Interior Eigenvalues:

$$\widetilde{K}^{-1}\widetilde{A_\theta}t = -\widetilde{K}^{-1}r$$

where $\widetilde{A_\theta} = (I - uu^*)(A - \theta I)(I - uu^*)$ and $\widetilde{K} \approx \widetilde{A_\tau}$

Exterior Eigenvalues - Arithmetic

- One complex (JD-Complex) version and 3 real versions ($JD-P_0$, $JD-P_1$ and $JD-P_2$)
- Residual:

$$[r_1\ r_2] = A[u_1\ u_2] - [u_1\ u_2] \begin{bmatrix} \Re(\theta) & \Im(\theta) \\ -\Im(\theta) & \Re(\theta) \end{bmatrix}. \quad (1)$$

- Correction Equation:

$$P \begin{bmatrix} A-\Re(\theta)I & \Im(\theta)I \\ -\Im(\theta)I & A-\Re(\theta)I \end{bmatrix} P \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \quad (2)$$

Exterior Eigenvalues - Arithmetic

- One complex (JD-Complex) version and 3 real versions ($JD - P_0$, $JD - P_1$ and $JD - P_2$)
- Residual:

$$[r_1 \ r_2] = A[u_1 \ u_2] - [u_1 \ u_2] \begin{bmatrix} \Re(\theta) & \Im(\theta) \\ -\Im(\theta) & \Re(\theta) \end{bmatrix}. \quad (1)$$

- Correction Equation:

$$P \begin{bmatrix} A - \Re(\theta) I & \Im(\theta) I \\ -\Im(\theta) I & A - \Re(\theta) I \end{bmatrix} P \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \quad (2)$$

Exterior Eigenvalues - Arithmetic

- One complex (JD-Complex) version and 3 real versions ($JD-P_0$, $JD-P_1$ and $JD-P_2$)
- Residual:

$$[r_1\ r_2] = A[u_1\ u_2] - [u_1\ u_2]\begin{bmatrix} \Re(\theta) & \Im(\theta) \\ -\Im(\theta) & \Re(\theta) \end{bmatrix}. \qquad (1)$$

- Correction Equation:

$$P\begin{bmatrix} A-\Re(\theta)I & \Im(\theta)I \\ -\Im(\theta)I & A-\Re(\theta)I \end{bmatrix}P\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = -\begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \qquad (2)$$

Exterior Eigenvalues - Arithmetic

- One complex (JD-Complex) version and 3 real versions ($JD-P_0$, $JD-P_1$ and $JD-P_2$)
- Residual:

$$[r_1\ r_2] = A[u_1\ u_2] - [u_1\ u_2] \begin{bmatrix} \Re(\theta) & \Im(\theta) \\ -\Im(\theta) & \Re(\theta) \end{bmatrix}. \qquad (1)$$

- Correction Equation:

$$P \begin{bmatrix} A - \Re(\theta)I & \Im(\theta)I \\ -\Im(\theta)I & A - \Re(\theta)I \end{bmatrix} P \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \qquad (2)$$

Exterior Eigenvalues - Projectors in Real Arithmetic

Projector P_0 ($JD - P_0$):

$$P_0 = I_{2n} - \begin{bmatrix} u_1 & u_2 \\ u_2 & -u_1 \end{bmatrix} \begin{bmatrix} u_1^T & u_2^T \\ u_2^T & -u_1^T \end{bmatrix}$$

Projector P_1 ($JD - P_1$):

$$P_1 = I_{2n} - \begin{bmatrix} \hat{u}_1 & \hat{u}_2 & 0 & 0 \\ 0 & 0 & \hat{u}_1 & \hat{u}_2 \end{bmatrix} \begin{bmatrix} \hat{u}_1^T & 0 \\ \hat{u}_2^T & 0 \\ 0 & \hat{u}_1^T \\ 0 & \hat{u}_2^T \end{bmatrix}$$

Projector P_2 ($JD - P_2$):

$$P_2 = I_{2n} - \begin{bmatrix} \hat{u}_1 & 0 \\ 0 & \hat{u}_2 \end{bmatrix} \begin{bmatrix} \hat{u}_1^T & 0 \\ 0 & \hat{u}_2^T \end{bmatrix}$$

Exterior Eigenvalues - Projectors in Real Arithmetic

Projector P_0 ($JD - P_0$):

$$P_0 = I_{2n} - \begin{bmatrix} u_1 & u_2 \\ u_2 & -u_1 \end{bmatrix} \begin{bmatrix} u_1^T & u_2^T \\ u_2^T & -u_1^T \end{bmatrix}$$

Projector P_1 ($JD - P_1$):

$$P_1 = I_{2n} - \begin{bmatrix} \hat{u}_1 & \hat{u}_2 & 0 & 0 \\ 0 & 0 & \hat{u}_1 & \hat{u}_2 \end{bmatrix} \begin{bmatrix} \hat{u}_1^T & 0 \\ \hat{u}_2^T & 0 \\ 0 & \hat{u}_1^T \\ 0 & \hat{u}_2^T \end{bmatrix}$$

Projector P_2 ($JD - P_2$):

$$P_2 = I_{2n} - \begin{bmatrix} \hat{u}_1 & 0 \\ 0 & \hat{u}_2 \end{bmatrix} \begin{bmatrix} \hat{u}_1^T & 0 \\ 0 & \hat{u}_2^T \end{bmatrix}$$

Exterior Eigenvalues - Projectors in Real Arithmetic

Projector P_0 ($JD - P_0$):

$$P_0 = I_{2n} - \begin{bmatrix} u_1 & u_2 \\ u_2 & -u_1 \end{bmatrix} \begin{bmatrix} u_1^T & u_2^T \\ u_2^T & -u_1^T \end{bmatrix}$$

Projector P_1 ($JD - P_1$):

$$P_1 = I_{2n} - \begin{bmatrix} \hat{u}_1 & \hat{u}_2 & 0 & 0 \\ 0 & 0 & \hat{u}_1 & \hat{u}_2 \end{bmatrix} \begin{bmatrix} \hat{u}_1^T & 0 \\ \hat{u}_2^T & 0 \\ 0 & \hat{u}_1^T \\ 0 & \hat{u}_2^T \end{bmatrix}$$

Projector P_2 ($JD - P_2$):

$$P_2 = I_{2n} - \begin{bmatrix} \hat{u}_1 & 0 \\ 0 & \hat{u}_2 \end{bmatrix} \begin{bmatrix} \hat{u}_1^T & 0 \\ 0 & \hat{u}_2^T \end{bmatrix}$$

Exterior Eigenvalues - Projectors in Real Arithmetic

Projector P_0 ($JD - P_0$):

$$P_0 = I_{2n} - \begin{bmatrix} u_1 & u_2 \\ u_2 & -u_1 \end{bmatrix} \begin{bmatrix} u_1^T & u_2^T \\ u_2^T & -u_1^T \end{bmatrix}$$

Projector P_1 ($JD - P_1$):

$$P_1 = I_{2n} - \begin{bmatrix} \hat{u}_1 & \hat{u}_2 & 0 & 0 \\ 0 & 0 & \hat{u}_1 & \hat{u}_2 \end{bmatrix} \begin{bmatrix} \hat{u}_1^T & 0 \\ \hat{u}_2^T & 0 \\ 0 & \hat{u}_1^T \\ 0 & \hat{u}_2^T \end{bmatrix}$$

Projector P_2 ($JD - P_2$):

$$P_2 = I_{2n} - \begin{bmatrix} \hat{u}_1 & 0 \\ 0 & \hat{u}_2 \end{bmatrix} \begin{bmatrix} \hat{u}_1^T & 0 \\ 0 & \hat{u}_2^T \end{bmatrix}$$

Exterior Eigenvalues - Projectors in Real Arithmetic

- In theory, P_0 is the same as in the complex version.
- P_1 represents the (orthogonal) projector onto the orthogonal complement of the subspace $\mathcal{U} = \text{span}\{u_1, u_2\}$
- $\{\hat{u}_1, \hat{u}_2\}$ is an orthogonal basis of $\mathcal{U}$.
- $\{\hat{u}_1, \hat{u}_2\}$ is cheaply obtained from the Schur decomposition of the projected problem.
- P_2 represents a lighter projector.

Exterior Eigenvalues - Projectors in Real Arithmetic

- In theory, P_0 is the same as in the complex version.
- P_1 represents the (orthogonal) projector onto the orthogonal complement of the subspace $\mathcal{U} = \text{span}\{u_1, u_2\}$
- $\{\hat{u}_1, \hat{u}_2\}$ is an orthogonal basis of $\mathcal{U}$.
- $\{\hat{u}_1, \hat{u}_2\}$ is cheaply obtained from the Schur decomposition of the projected problem.
- P_2 represents a lighter projector.

Exterior Eigenvalues - Projectors in Real Arithmetic

- In theory, P_0 is the same as in the complex version.
- P_1 represents the (orthogonal) projector onto the orthogonal complement of the subspace $\mathcal{U} = \text{span}\{u_1, u_2\}$
- $\{\hat{u}_1, \hat{u}_2\}$ is an orthogonal basis of $\mathcal{U}$.
- $\{\hat{u}_1, \hat{u}_2\}$ is cheaply obtained from the Schur decomposition of the projected problem.
- P_2 represents a lighter projector.

Exterior Eigenvalues - Projectors in Real Arithmetic

- In theory, P_0 is the same as in the complex version.
- P_1 represents the (orthogonal) projector onto the orthogonal complement of the subspace $\mathcal{U} = \text{span}\{u_1, u_2\}$
- $\{\hat{u}_1, \hat{u}_2\}$ is an orthogonal basis of $\mathcal{U}$.
- $\{\hat{u}_1, \hat{u}_2\}$ is cheaply obtained from the Schur decomposition of the projected problem.
- P_2 represents a lighter projector.

Interior Eigenvalues

Algorithm 2 Harmonic extraction in Jacobi-Davidson algorithm

$w \leftarrow (A - \tau I)v_m$, where v_m is the last column of V

$W \leftarrow$ Orthonormalize$[W, w]$, with h the orthogonalization coefficients and ρ the norm prior to normalization

Update $N \leftarrow \begin{bmatrix} N & h \\ 0 & \rho \end{bmatrix}$,

Update M such that $M = W^* V$

Compute generalized Schur decomposition $NQ = Z\tilde{N}$, $MQ = Z\tilde{M}$ such that $|\tilde{n}_{1,1}/\tilde{m}_{1,1}|$ is the minimum

$s \leftarrow q_1$ and $\theta \leftarrow \overline{\tilde{m}}_{1,1}\tilde{n}_{1,1} + \tau$

Interior Eigenvalues

- Harmonic-Ritz approach leads to:

JD-pre:

$$V^*(A-\tau I)^*(A-\tau I)Vs = (\theta-\tau)V^*(A-\tau I)^*Vs$$

JD-rev:

$$V^*(A-\tau I)^*Vs = \frac{1}{\theta-\tau}V^*(A-\tau I)^*(A-\tau I)Vs$$

- Preconditioners tested for the C.E.: $\{Jacobi,\ ILU(0),\ LU\}$

Interior Eigenvalues

- Harmonic-Ritz approach leads to:

JD-pre:

$$V^*(A-\tau I)^*(A-\tau I)Vs = (\theta-\tau)V^*(A-\tau I)^*Vs$$

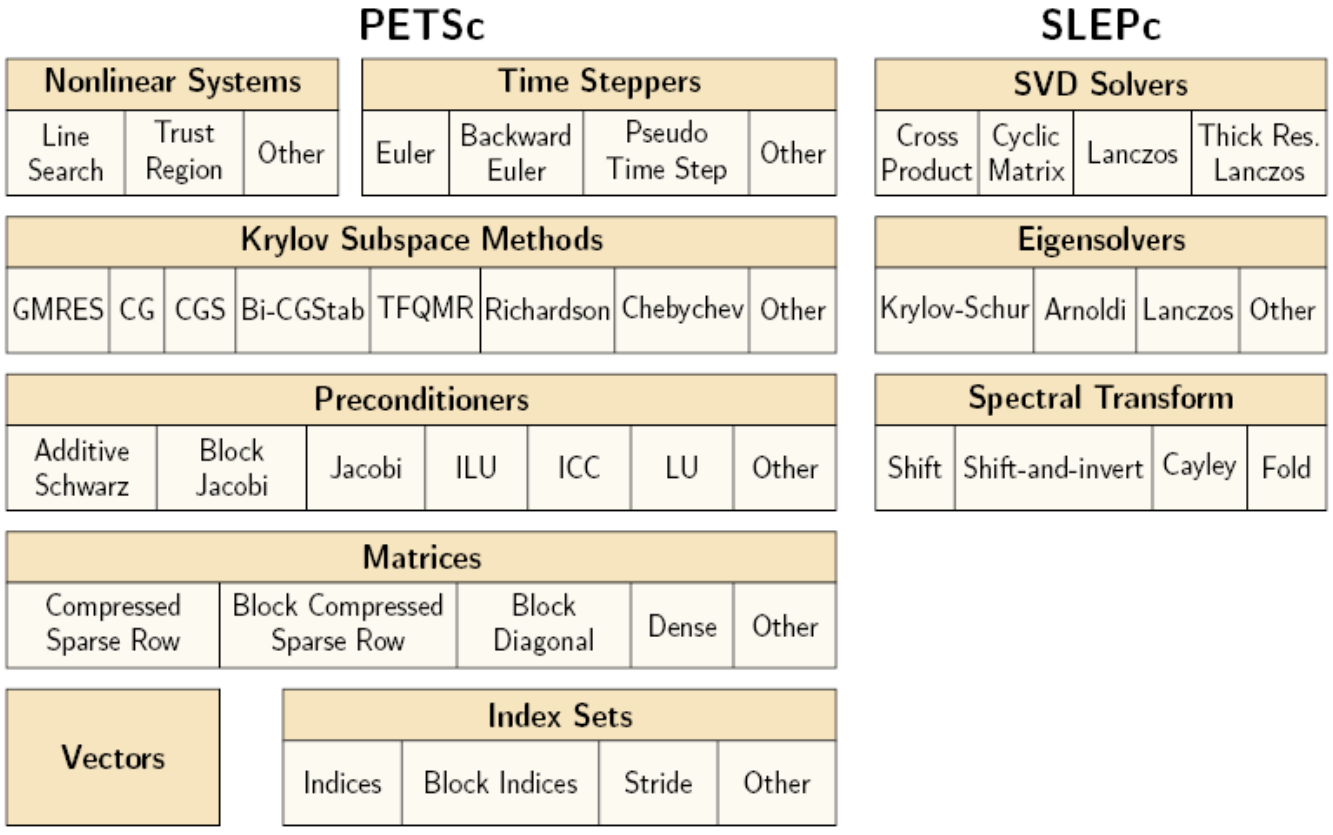
JD-rev:

$$V^*(A-\tau I)^*Vs = \frac{1}{\theta-\tau}V^*(A-\tau I)^*(A-\tau I)Vs$$

- Preconditioners tested for the C.E.: $\{Jacobi,\ ILU(0),\ LU\}$

The PETSc/SLEPc framework

PETSc: Portable, Extensible Toolkit for Scientific Computation
SLEPc: Scalable Library for Eigenvalue Problem Computations

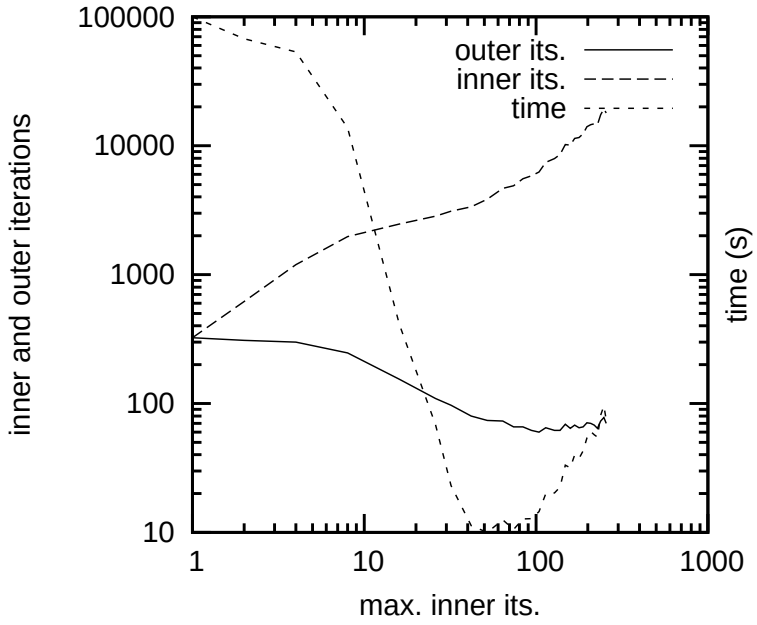


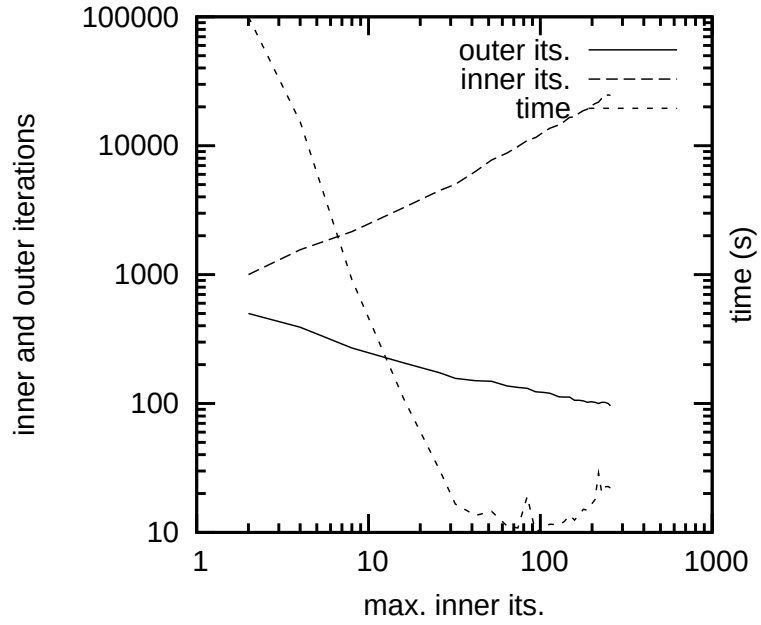
Additional details

- $u_0 = A^k \times rand(n, 1)$, (default, $k = 5$)
- Projected eigensystem solution:
  1. Exterior eigenvalues: LAPACK _GEHRD and _HSEQR via SLEPc
  2. Interior eigenvalues: LAPACK _GGES
- Orthogonalization: Classical Gram-Schmidt with reorthogonalization.

Searching for a “default”

- **Goals:** Robustness and “low” computation time.





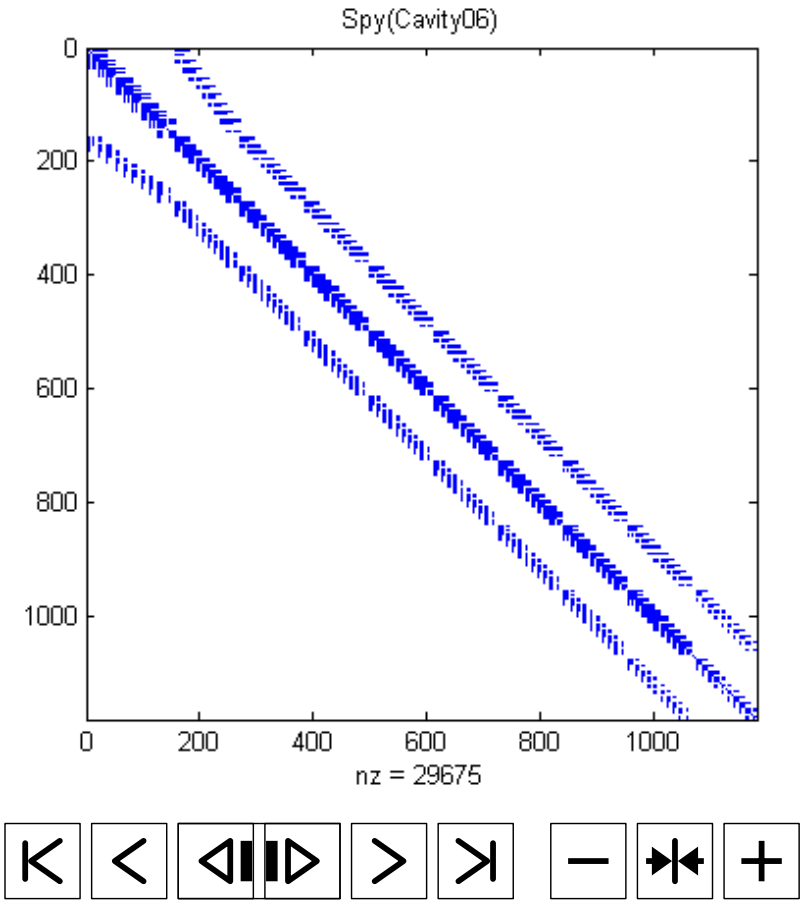
- Low demandings (*tol* and *itmax*) for inner iteration.

Hardware

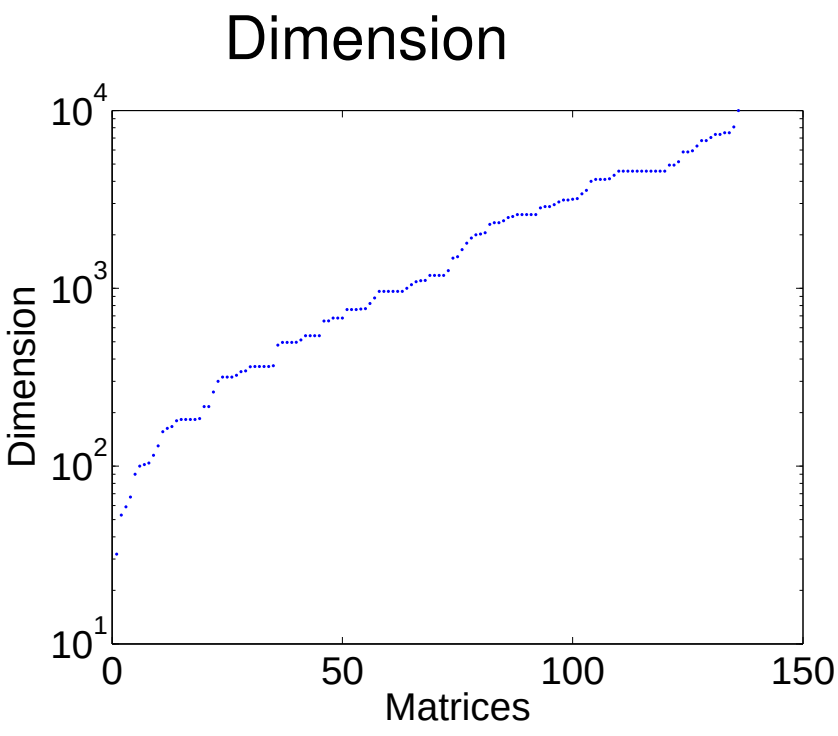
1. Odin - Universidad Politécnica de Valencia
   - Cluster Pentium Xeon
   - 55 dual-processor nodes
   - 2.8 GHz and 1 GB per node
   - High-speed SCI network with 2D torus topology
2. CaesarAugusta - Barcelona Supercomputing Center
   - 64-bit PowerPC 970FX processors
   - 256 JS20 blade computing nodes
   - 2.2 GHz
   - Low latency Myrinet network

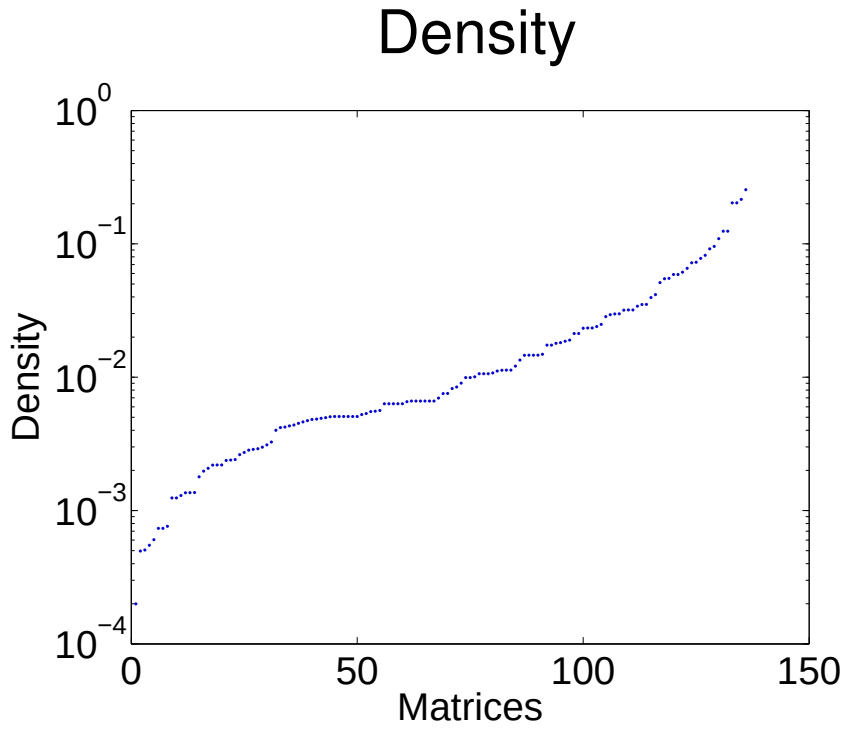
Testbed

136 Matrices from Matrix Market and UF Sparse Matrix Collection

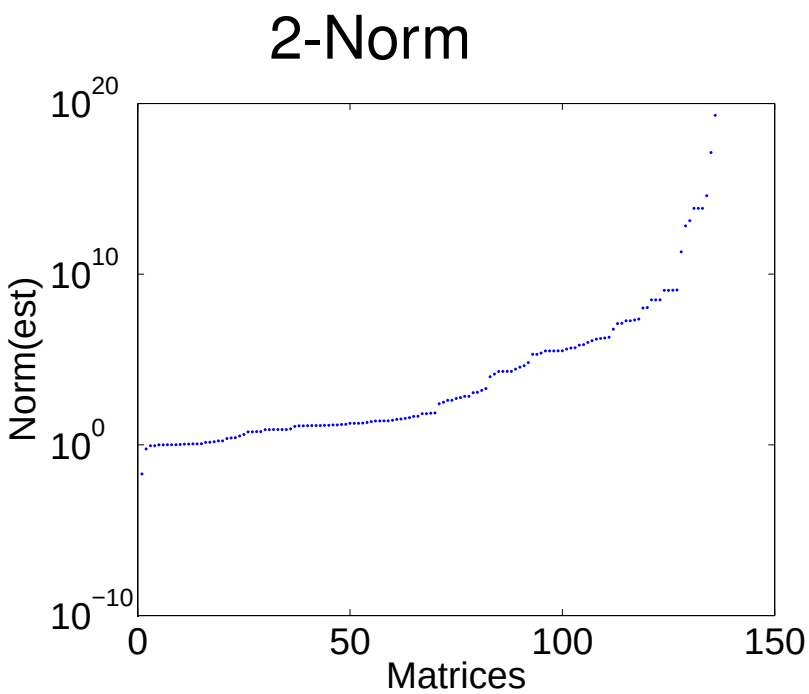


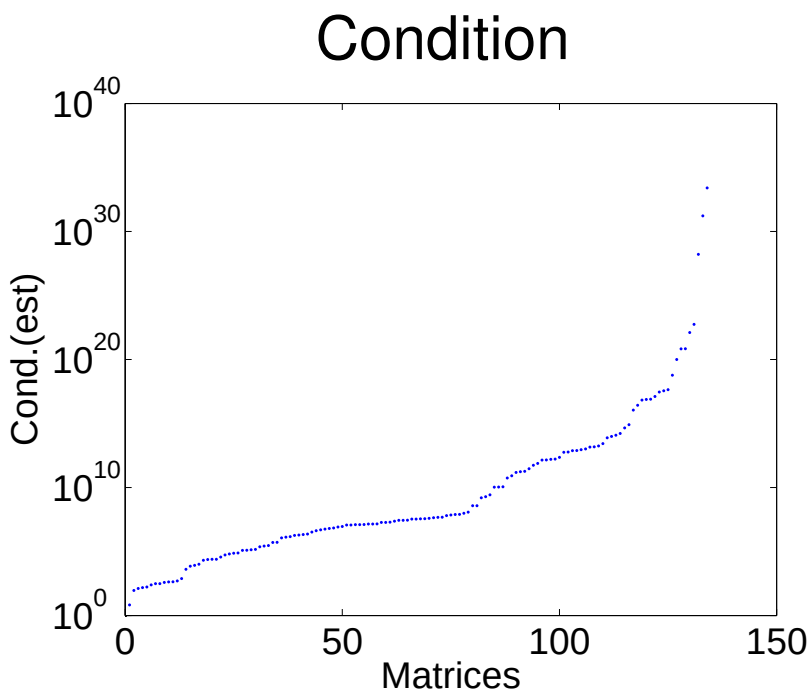
Testbed


Dimension
Dimension
10^4
10^3
10^2
10^1
0
50
100
150
Matrices


Density
Density
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
0
50
100
150
Matrices

Testbed


2-Norm
Norm(est)
10^{20}
10^{10}
10^{0}
10^{-10}
0
50
100
150
Matrices


Condition
Cond.(est)
10^{40}
10^{30}
10^{20}
10^{10}
10^{0}
0
50
100
150
Matrices

Parameters

Exterior

- 4 vers.(arithm.):
  1 complex and 3 real
- Power its: $\{0, 5\}$
- Max. Inner its: $\{5, 20, 40\}$
- Inner tol.: 10^{-3}
- Max. Outer its: 500
- Outer tol.: 10^{-7}
- Nr. Tests: 3264
- Best Sucess Rate: 100%

Interior

- 4 precond.:
  $\{ILU, Jacobi, LU, none\}$
- Target: $\tau = 0$
- Max. Inner its: $\{50\}$
- Inner tol.: 10^{-3}
- Max. Outer its: 500
- Outer tol.: 10^{-7}
- Nr. Tests: 1088
- Best Sucess Rate: 81%

Wilcoxon non-parametric test: $\alpha = 5\%$

Parameters

Exterior

- 4 vers.(arithm.):
  1 complex and 3 real
- Power its: $\{0, 5\}$
- Max. Inner its: $\{5, 20, 40\}$
- Inner tol.: 10^{-3}
- Max. Outer its: 500
- Outer tol.: 10^{-7}
- Nr. Tests: 3264
- Best Sucess Rate: 100%

Interior

- 4 precond.:
  $\{ILU, Jacobi, LU, none\}$
- Target: $\tau = 0$
- Max. Inner its: $\{50\}$
- Inner tol.: 10^{-3}
- Max. Outer its: 500
- Outer tol.: 10^{-7}
- Nr. Tests: 1088
- Best Sucess Rate: 81%

Wilcoxon non-parametric test: $\alpha = 5\%$

Overview

Exterior eigenvalues:

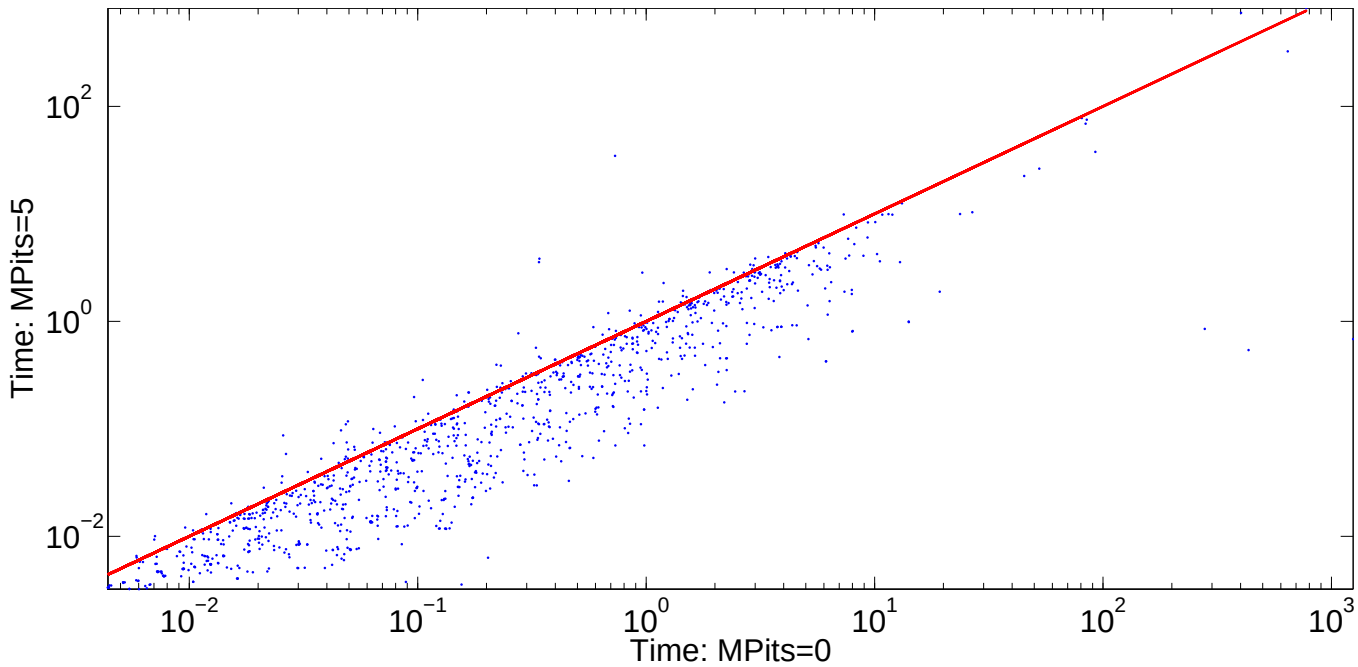
PowerIts	0			5		
LinearIts	5	20	40	5	20	40
$JD - Complex$	134	135	135	134	136	136
$JD - P_0$	134	96	103	118	129	129
$JD - P_1$	132	133	132	133	133	134
$JD - P_2$	134	133	134	134	133	134

Interior eigenvalues:

Precond	$ILU(0)$	*Jacobi*	*LU*	–
$jdh - pre$	56	64	104	76
$jdh - rev$	57	67	110	84

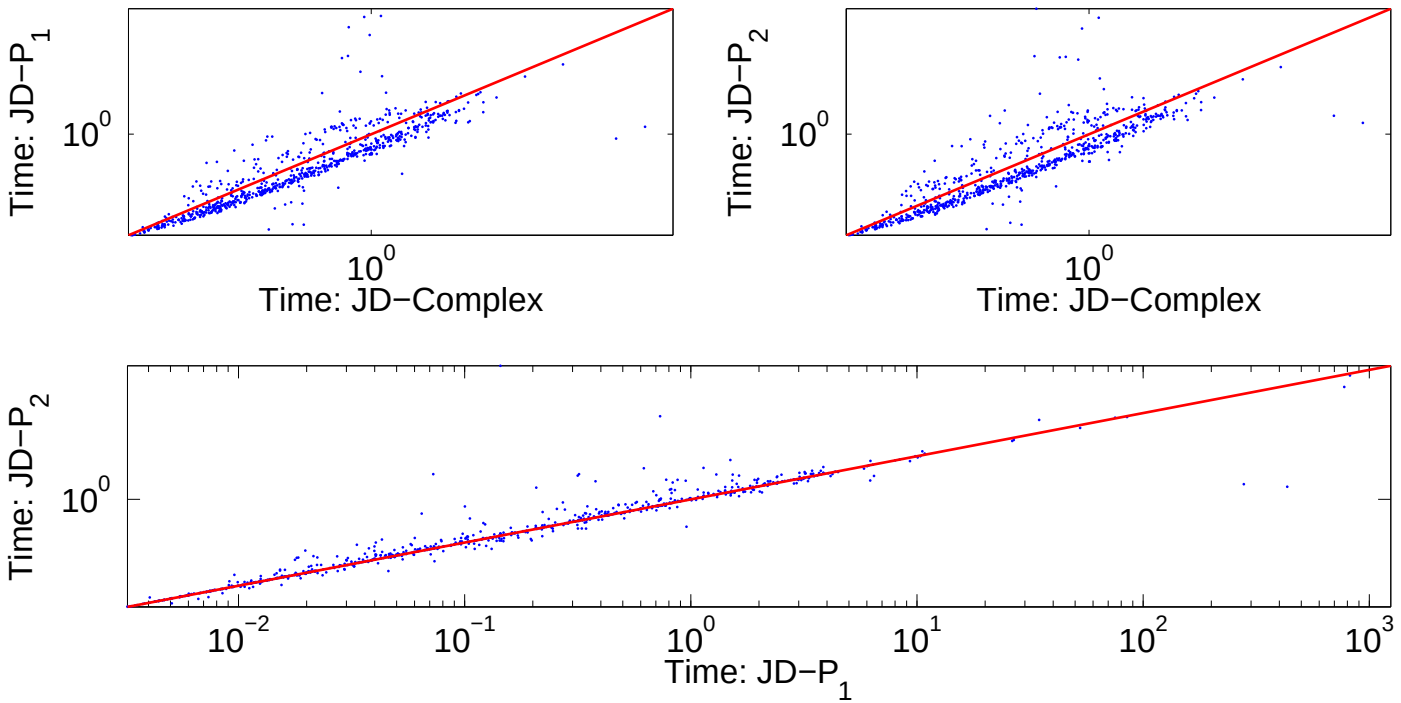
Power Iterations

- More matrices and less time.



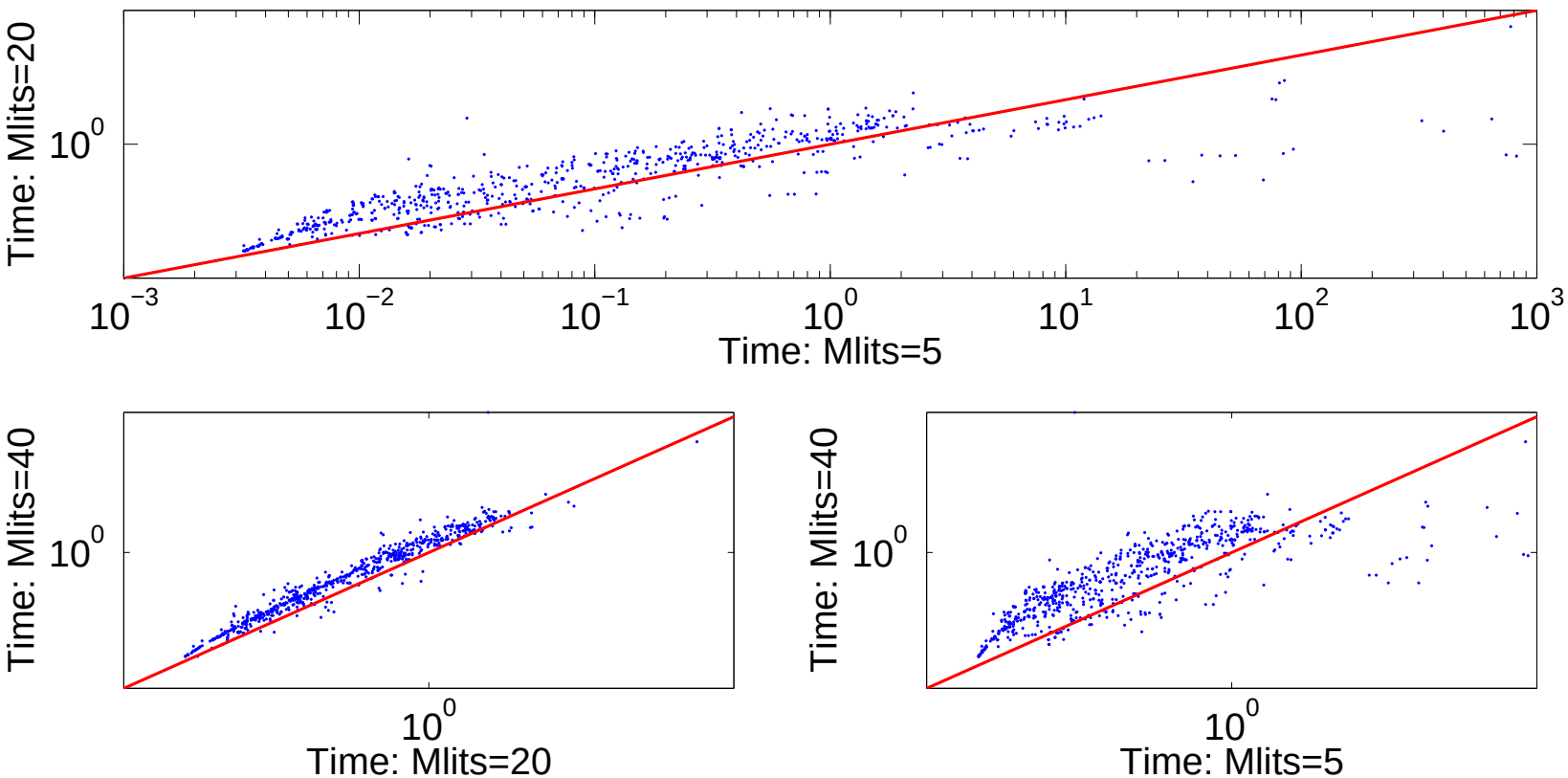
Arithmetic

- Computation Time: Real ($P_1 \approx P_2$) better than Complex.



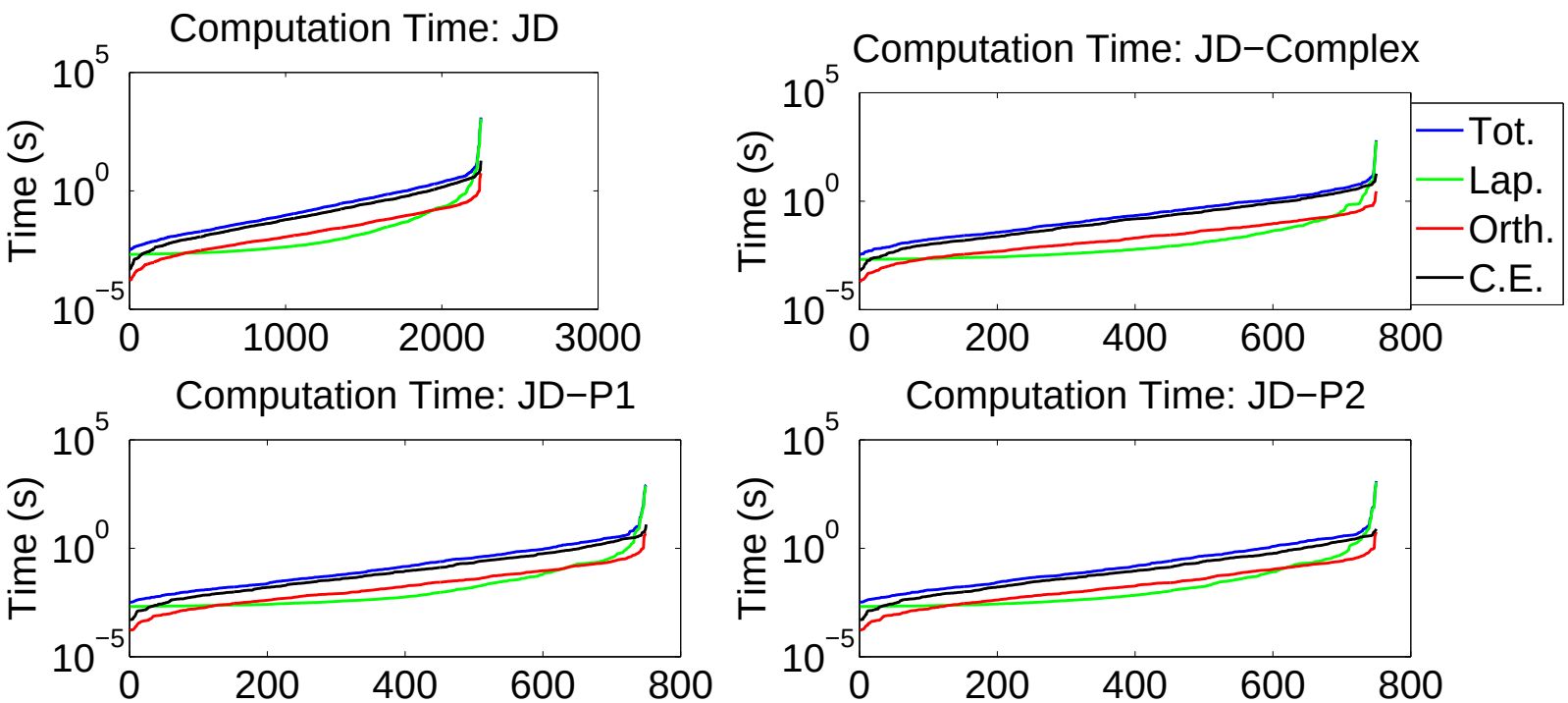
Correction Equation

- Computation time: $T_5 \leq T_{20} \ll T_{40}$.

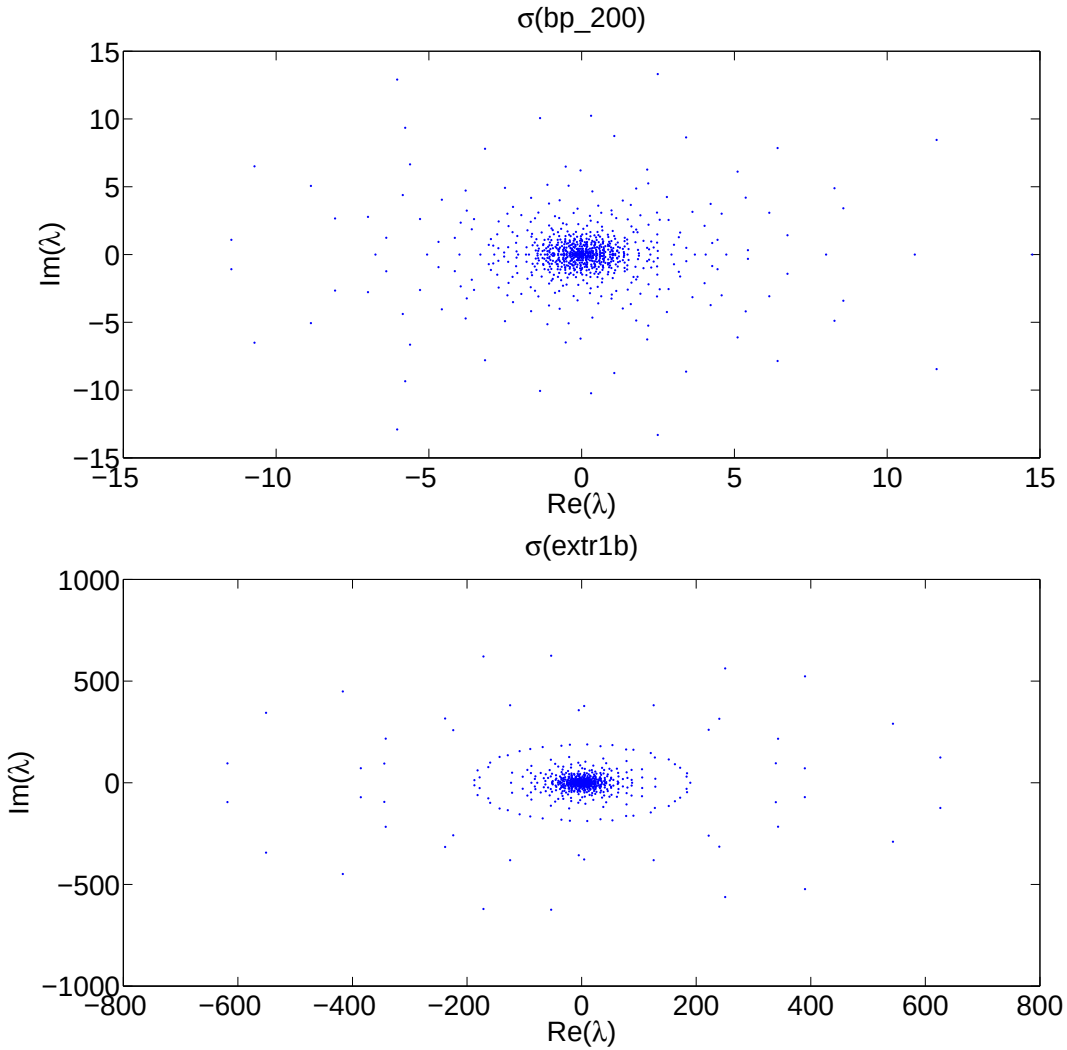


Computation time distribution: main phases

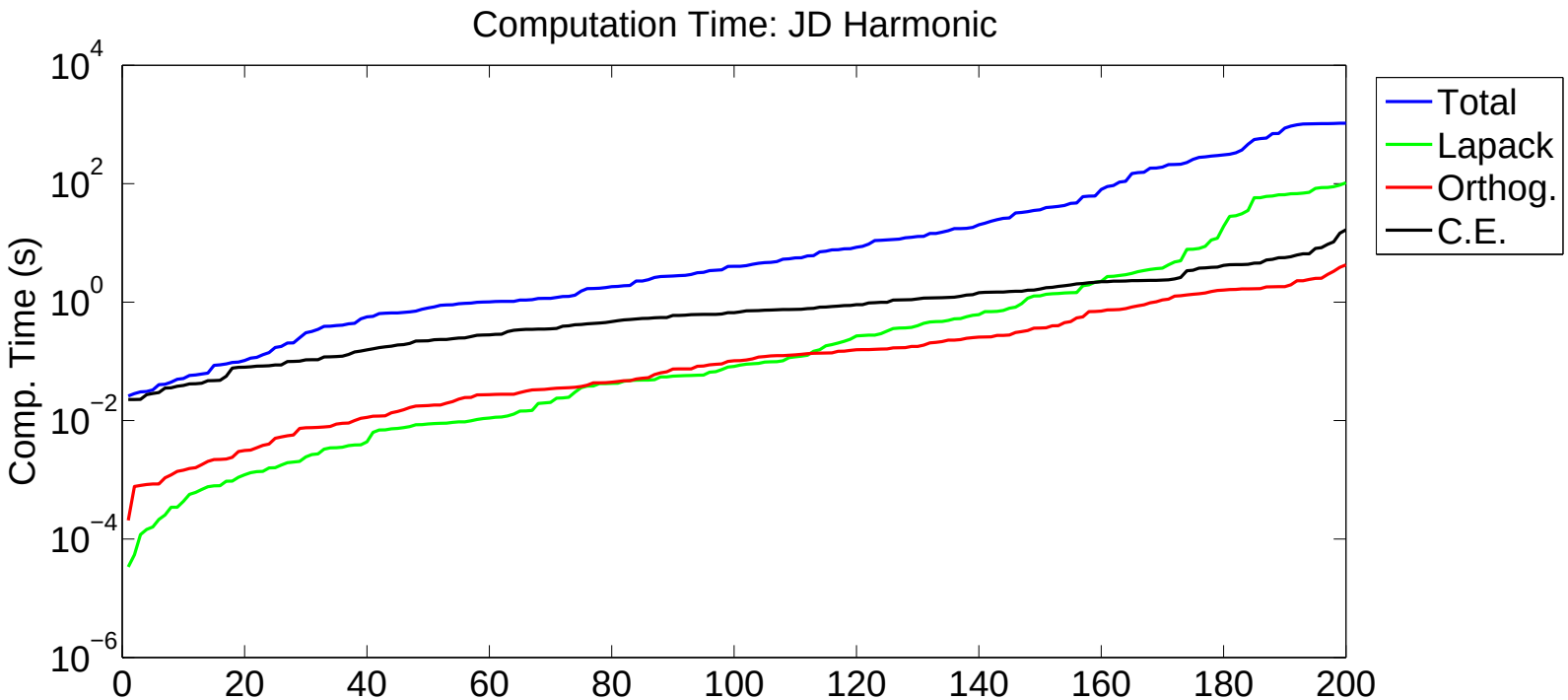
- Arithmetic non-dependent.



The Interior Case: some difficulties

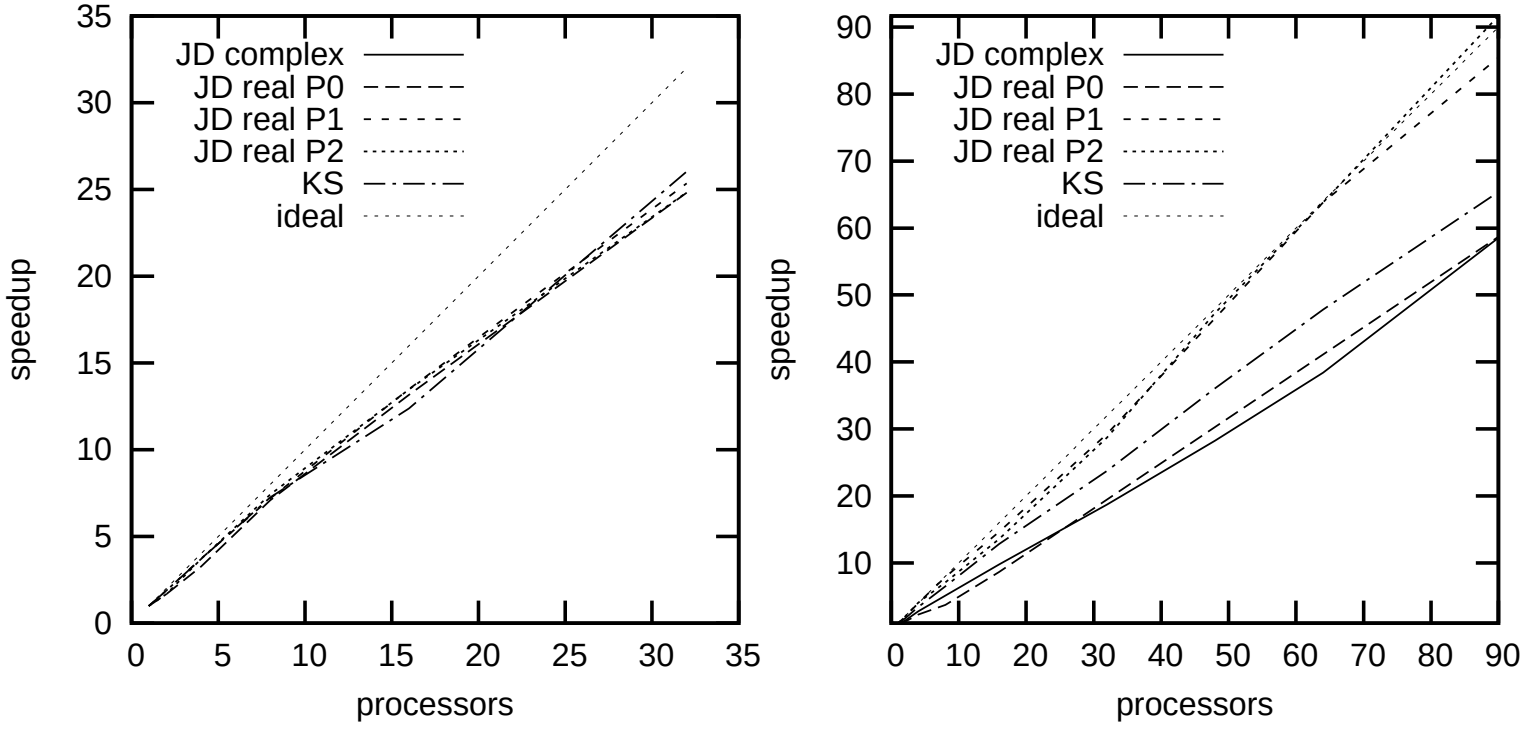


Computation time distribution: main phases



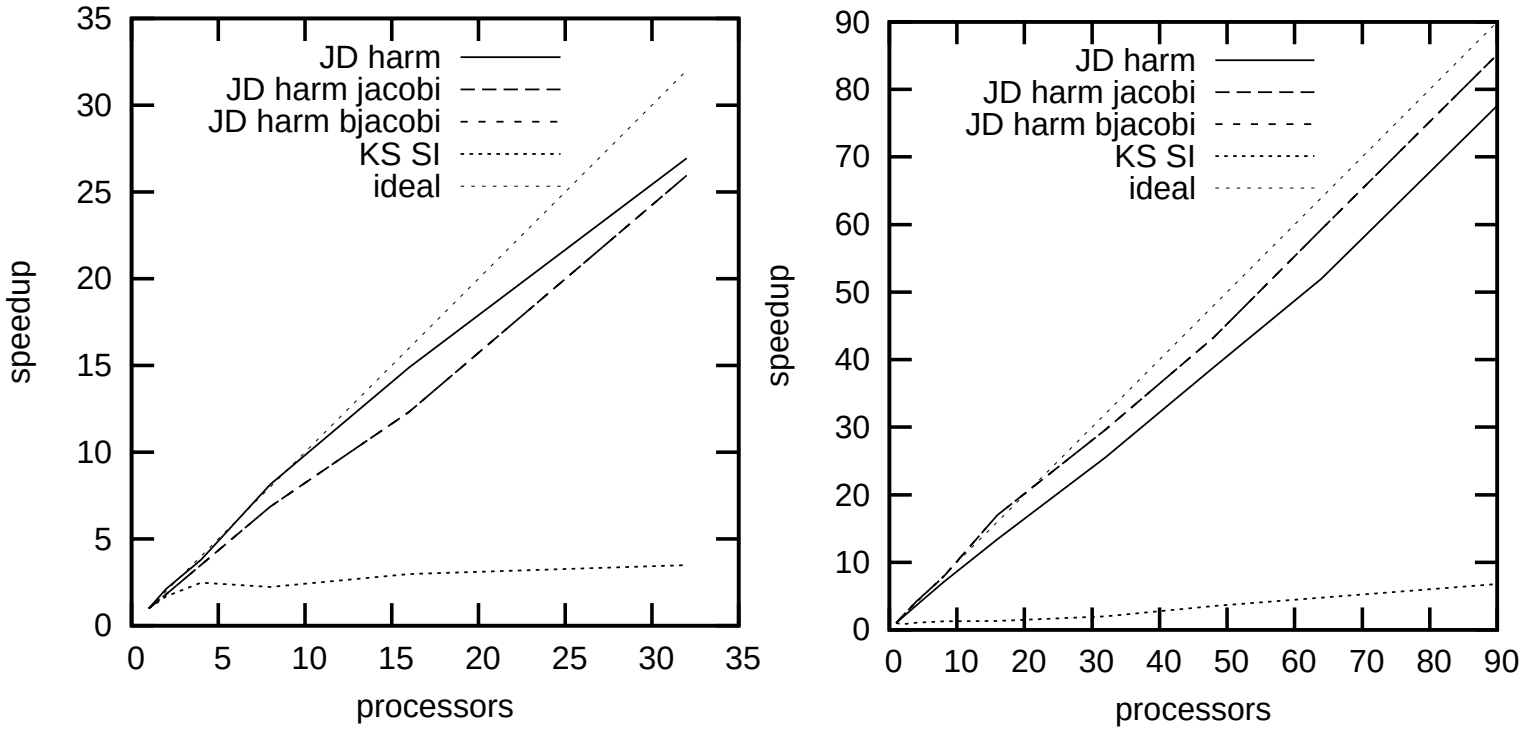
Speedup's: *tmt_unsym*
$n = 917,825 \quad nnz = 4,584,801$

Speedup on Odin (left) and CaesarAugusta (right)



Speedup's: *tmt_unsym* ($\tau = 0$)

Speedup on Odin (left) and CaesarAugusta (right)



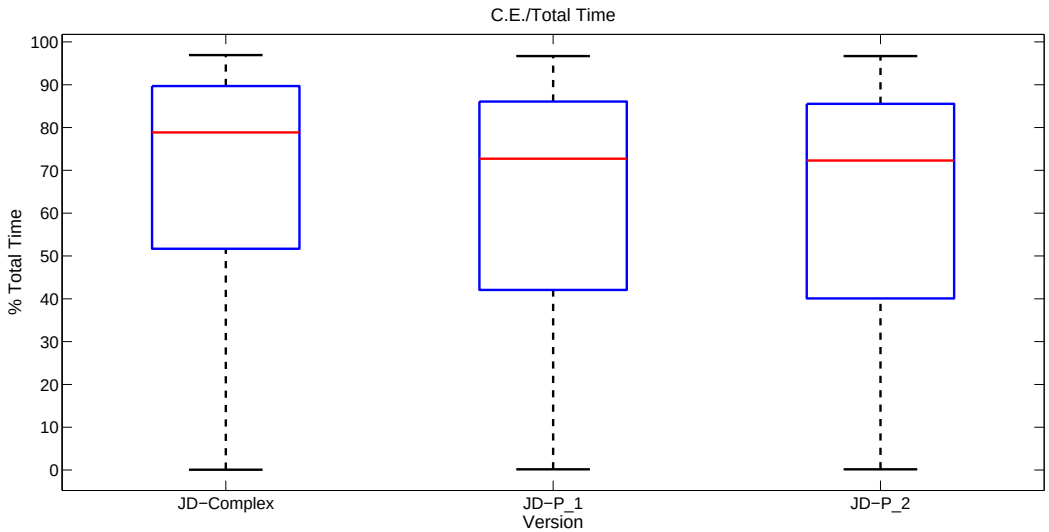
Conclusions and future work

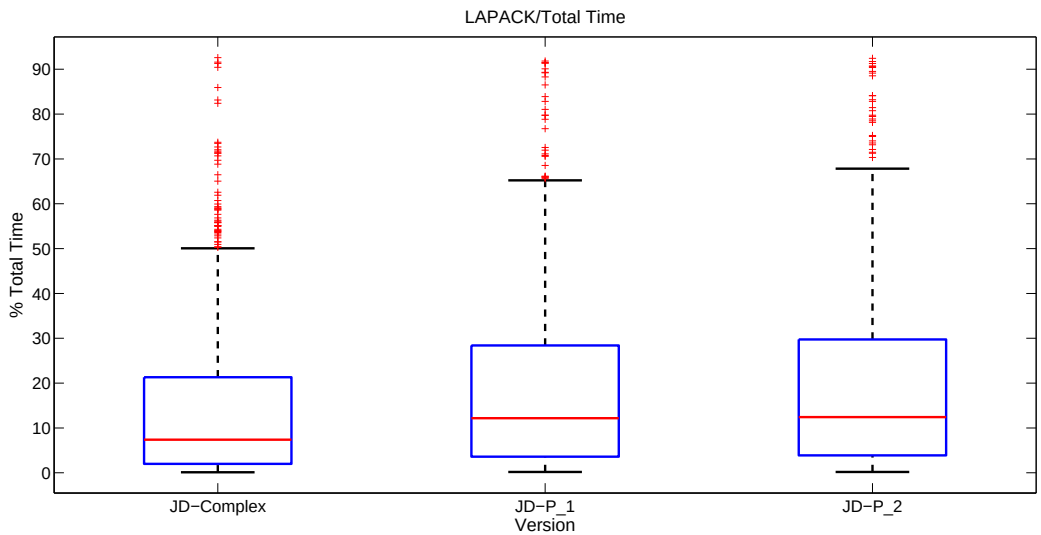
- A parallel implementation of the JD method for unsymmetric matrices was implemented with complex and real arithmetic versions.
- The code is already able to compute exterior as well as interior eigenvalues with a reasonable efficiency and showing good speedups.

- The project is been developed, improving the code and adding new features in order to deliver a better solution.
- Restart and locking of converged eigenpairs is being developed by SLEPc team
  - A first approach is been tested and used on an application in a plasma turbulence code.
- A more elaborate adaptive stopping criterion for the C.E. solver may be useful.
- These developments will be added to a fully featured JD solver in SLEPc.

Computation time dist: proj. problem vs C.E.

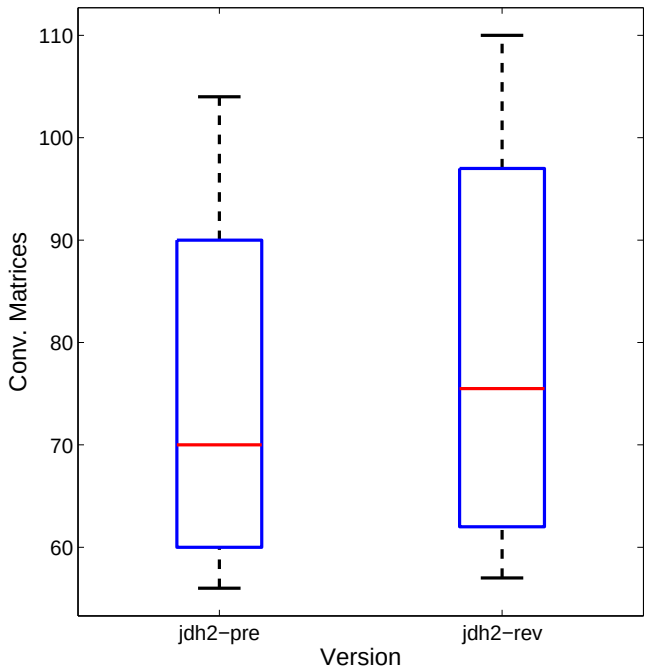
Computation time dist: proj. problem vs C.E.

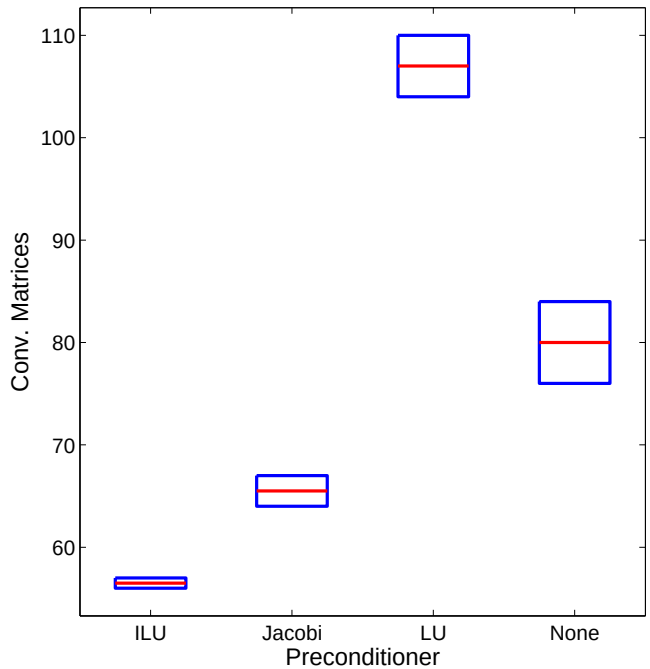


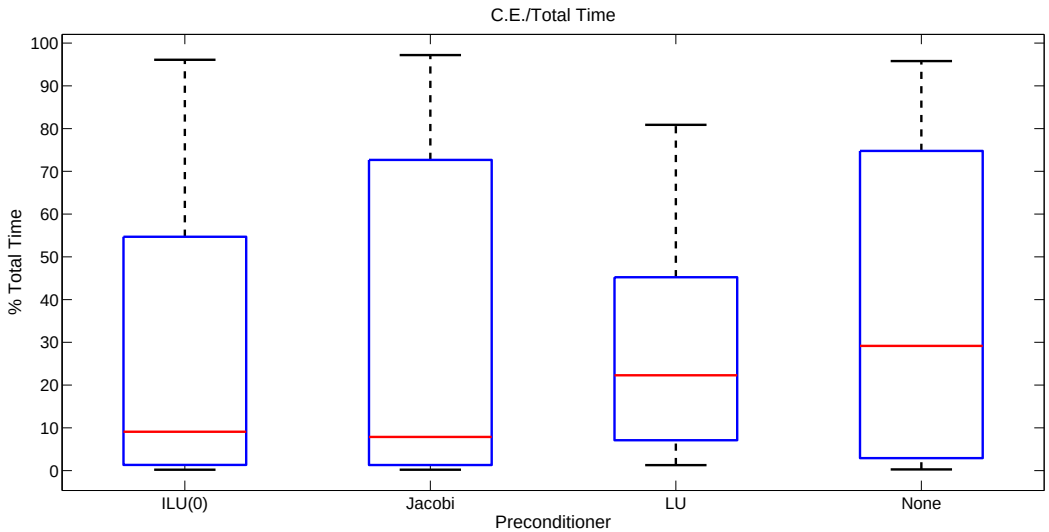
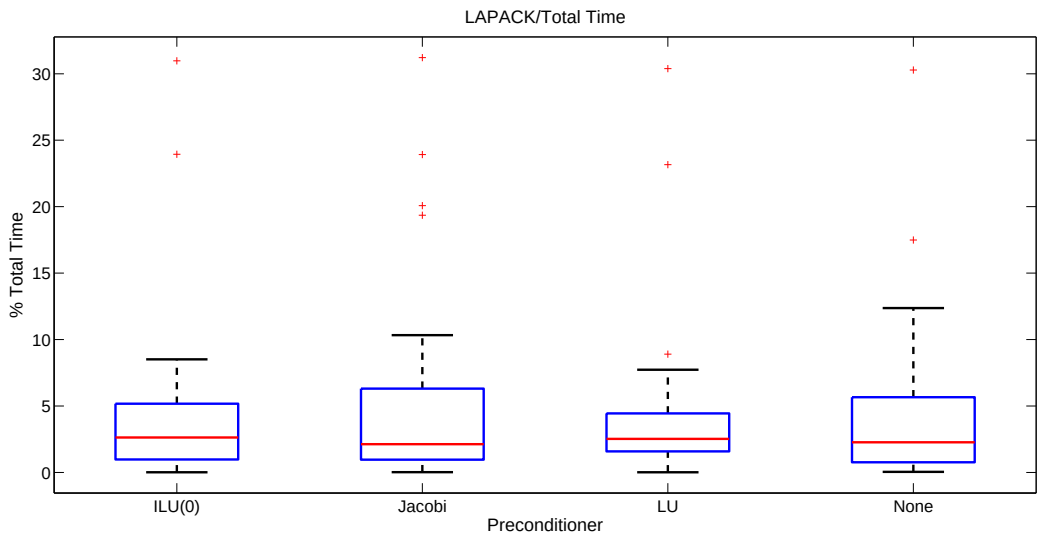


Overview: Versions and Preconditioners.

Overview: Versions and Preconditioners.





Computation time distribution: main phases

Computation time distribution: main phases