A Comparison of Eigensolvers for Large-Scale Three-Dimensional Problems

U. Hetmaniuk

Computational Math and Algorithms Dept.

Sandia National Laboratories

March 31, 2004

Joint work with P. Arbenz (ETH Zürich) and R. Lehoucq (SNL)

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Sandia National Laboratories

- **Funded by US Department of Energy**
- **Located Albuquerque (NM) & Livermore (CA)**
- **8,300 people**
- **Budget: $2 billion per year**
- **Mission of stockpile stewardship**
- **Capabilities:**
  - Biosciences
  - Chemical and Earth sciences
  - Electronics
  - Computer information
  - Nanotechnology
  - …

Sandia National Laboratories

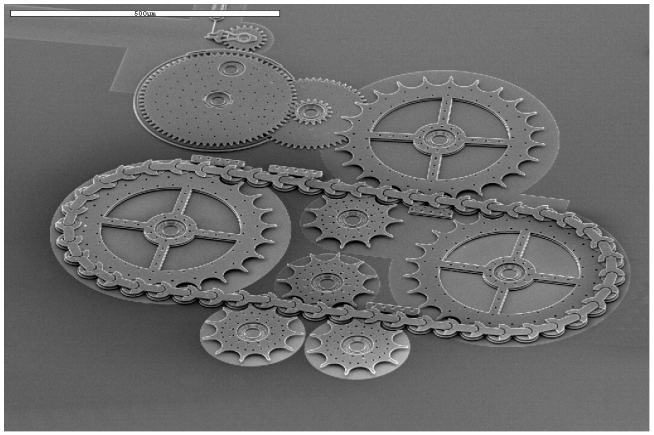
Achievements

Z machine:

- **Achieved an output of 80 times the entire world's output of electricity.**

MEMS:

World's smallest microchain drive



Asci Red:

- **First Tflop system (1.8 Tflops)**
- **9,152 Pentium Pro processors**
- **128 MB per processor**

Sandia National Laboratories

Computer Science & Mathematics

- **Development of advanced computing architecture, network, and facilities**
  - ASCI Red: Teraflop system.
  - Cplant: Commodity-based large-scale computing.
- **Software frameworks and solutions for high-performance distributed computing**
  - Trilinos: A collection of solver components.
  - Zoltan: Data management services for parallel applications.
- **Computational methods and codes for selected science and engineering simulations**
  - Dakota: A multilevel parallel, object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis.
  - ALEGRA: A three-dimensional, multi-material, arbitrary-lagrangian-eulerian code for solid dynamics.
  - MPSALSA: Massively parallel numerical methods for advanced simulation of chemically reacting flows.
- **Web page: http://www.cs.sandia.gov/**

Sandia National Laboratories

CSRI
Computer Science Research Institute

Research in computer science, computational science and mathematics through collaboration with university faculty, students and Sandia staff.

Impact

Research addressing Sandia problems in modeling and simulation
Recruiting through student programs
Improved quality and integration of Sandia research
Increased impact of Sandia research and codes.

Opportunities

Sabbaticals and summer faculty positions,
Postdocs and summer student positions,
Graduate fellowships,
Sponsored workshops, and
Technical visits and colloquia

Sandia National Laboratories

Contact: David E. Womble, dewombl@sandia.gov

- **Trilinos is an evolving framework which:**
  - efficiently reuses existing solver technology,
  - leverages new development across various projects,
  - satisfies specified practices for quality assurance.
- **Trilinos is a collection of *packages.***
- **Each package is:**
  - implemented in an object-oriented software framework.
  - focused on important and state-of-the-art algorithms in its problem regime.
  - developed by a small team of domain experts.
  - minimally dependent of parallel machine details.
  - minimally dependent on any other software packages (self-contained).
  - configurable / buildable / documented on its own (portability).
- **Open source: http://software.sandia.gov/trilinos**

Sandia National Laboratories

Primary Trilinos Packages

8/4/2003 - v12

- **Basic Linear Algebra Libraries**
  - Epetra: Current Production C++ Library
    - Epetra Core
    - Epetra Extensions
  - Tpetra: Next Generation C++ Library
  - Jpetra: Java Library
- **Common Services**
  - Teuchos: Parameter Lists, BLAS Interfaces, etc
- **Abstract Interfaces and Adaptors**
  - TSFCore: Basic Abstract classes
  - TSF Extensions: Aggregate/composite, overloaded operators
  - TSF Utilities: Core utility classes
- **Linear Solvers**
  - Amesos: OO Interfaces to 3rd party direct solvers
    - SuperLU
    - KundertSparse
    - SuperLUDist
    - DSCPack
    - UMFPack
    - MUMPS
  - AztecOO: Preconditioned Krylov Package based on Aztec
  - Komplex: Complex solver via equivalent real formulations
  - Belos: Next generation Krylov and block Krylov solvers
- **Preconditioners**
  - ML: Multi-level preconditioners
  - Meros: Segregated/Block Preconditioners
  - IFPACK: Algebraic preconditioners
- **Nonlinear Solvers**
  - NOX: Collection of nonlinear solvers
  - LOCA: Library of Continuation Algorithms
- **Eigensolvers**
  - Anasazi: Collection of eigensolvers
- **Time Integration**
  - TOX: Planned development
- **"New Package"**
  - "Hello World": Package Template to aid integration of new packages
  - Web site with layout and instructions

Sandia National Laboratories

Trilinos Packages

- **Epetra**
  - Concrete linear algebra classes (matrices, multivectors, graphs, operators, …).
  - Parallel code for linear algebra computations.
  - Portable interface to BLAS and LAPACK.
- **ML**
  - Multilevel preconditioning.
  - Geometric and algebraic multigrid.
- **AztecOO**
  - Based on Aztec.
    - Aztec is extracted from MPSalsa reacting flow code.
  - Algorithms: BiCGSTAB, CG, CGS, GMRES, TFQMR.
- **Anasazi**
  - Collection of eigensolvers

Problem

$$KQ = MQ\Lambda$$

- **K is a sparse and symmetric matrix *(stiffness).***
- **M is a sparse, symmetric, positive, definite matrix *(mass)*.**

  - **Frequency response in structural dynamics**
  - **Cavity analisys in electromagnetism**
  - **…**

Objective:

Determine the next generation approach to solve a generalized eigenproblem for extremely large 3D problems high into the frequency range

Sandia National Laboratories

Overview of Methods

- **Lanczos Method**
  - Boeing code — R. Grimes, J. Lewis, and H. Simon (1994)
  - Salinas code — 2002 Gordon Bell Prize Winner
- **Preconditioned Eigensolver**
  - The linear system $\mathbf{Ku} = \mathbf{f}$ is not solved accurately.
- **Component Mode Synthesis**
  - Over the last year, the AMLS method has become dominant for the frequency response analysis in the car industry.

Sandia National Laboratories

Lanczos Method

$$(K-\sigma M)^{-1}MV = VT + \sigma u e^{T}$$

- $V^{T}MV = I$.
- T is a tridiagonal matrix.
- Implicitly restarted Lanczos method (ARPACK).

Key computational issue: $x \rightarrow (K-\sigma M)^{-1}x$

- Boeing Code: Sparse factorization
- Salinas Code: Scalable FETI-DP solver (Farhat)

$$\sigma \leq 0$$

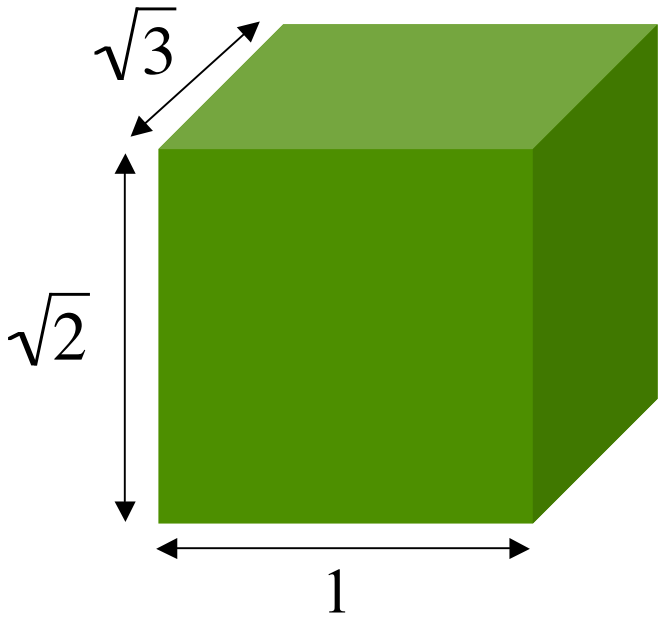
- Experiments: PCG with AMG preconditioner

$$\sigma = 0$$

Reference: R. Lehoucq, D. Sorensen, and C. Yang (1998)

Sandia National Laboratories

Model Problem



- We consider the Laplace equation with Dirichlet condition.
- The continuous eigenvalues can be multiple.
- The unit cube is discretized with 101 Q1 elements per direction.
- The smallest discrete eigenvalues are simple.

Platform and other details

- 16 processors (Dec Alpha processors).
- 512 MB memory per processor.
- CXML LAPACK and BLAS.
- All codes but ARPACK are in C++.
- The mass and stiffness matrix-vector multiplications are blocked (Epetra framework).
- The AMG preconditioner does not operate per block.

Post-processing checks:

- Mass-orthonormality of eigenvectors
- Missed eigenpairs
- Angles between the computed and discrete eigenspaces

Sandia National Laboratories

Convergence Check

Theoretical result:

$$|\lambda - \lambda_0| \leq \frac{\|Kx - Mx\lambda\|_{M^{-1}}}{\|x\|_M}$$

Convergence criterion:

$$\frac{1}{\sqrt{\lambda_1}} \frac{\|Kx - Mx\lambda\|_2}{\|x\|_M} \leq \varepsilon$$

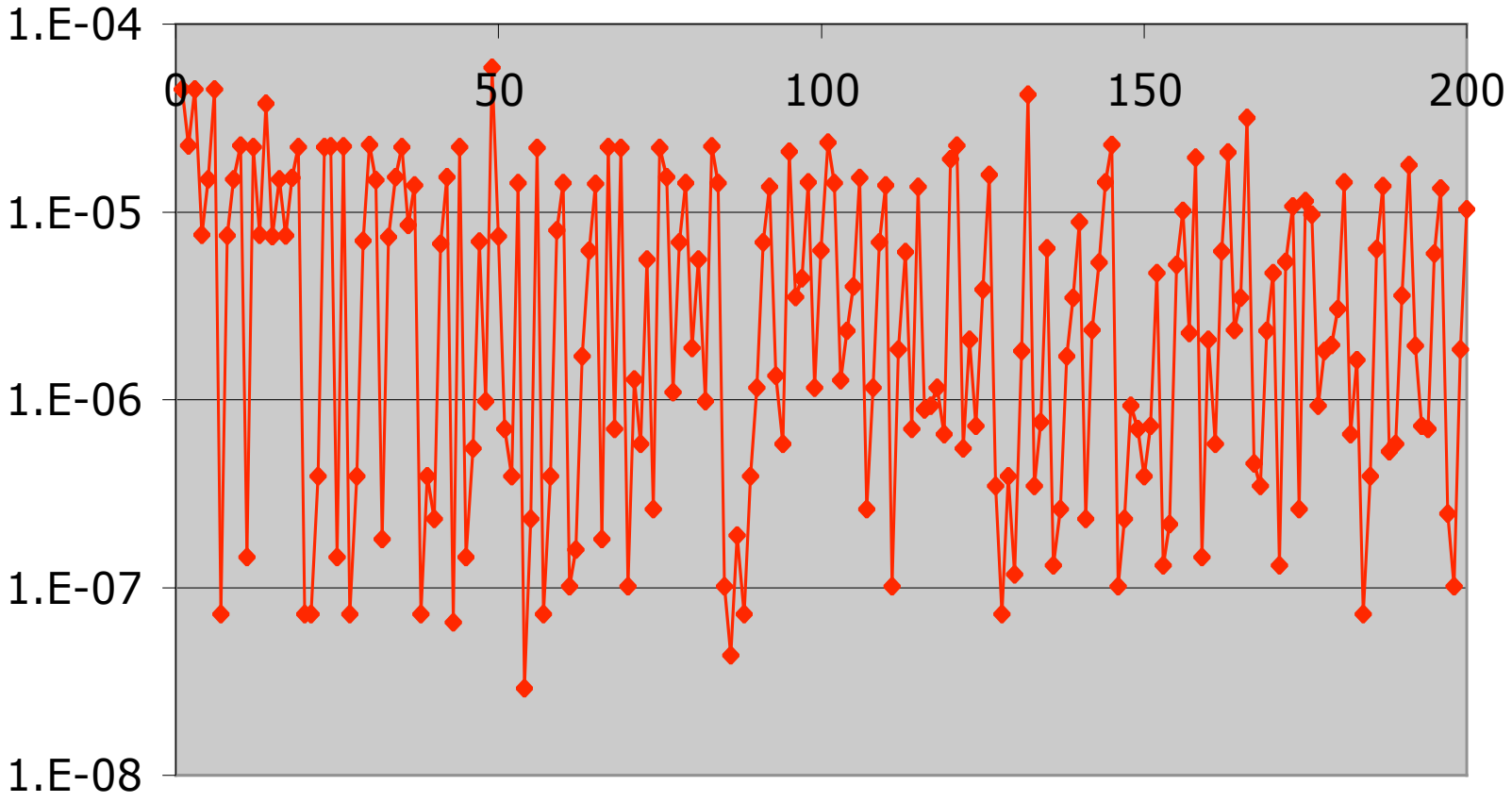
- λ_1 is the smallest eigenvalue of M
- ε is the tolerance ($10^{-4} \approx$ discretization error)

Sandia National Laboratories

Gap

$$\text{Gap} = \frac{\lambda_{j+1} - \lambda_j}{\lambda_{max} - \lambda_{min}}$$

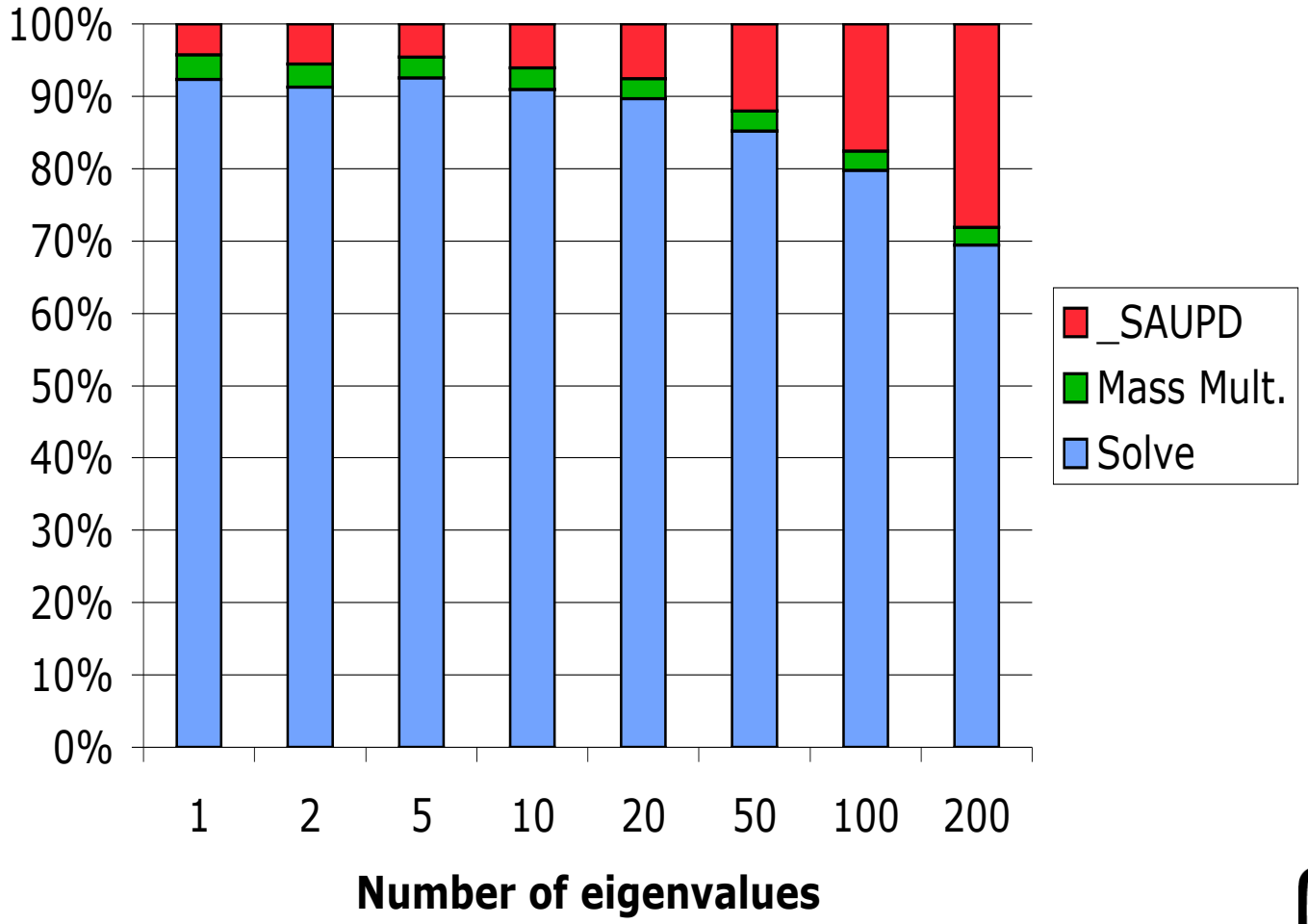
Spread ≈ $2*10^5$



Sandia National Laboratories

Time Distribution for ARPACK

- Number of eigenvalues requested: nev
- Size of working space: 2*nev



Sandia National Laboratories

Preconditioned Eigensolver

Idea: **Replace the operation $x \rightarrow (K-\sigma M)^{-1}x$ by $x \rightarrow N^{-1}x$ where N is a preconditioner for $K-\sigma M$**

Algorithms:

- **Gradient-based schemes: DACG & LOBPCG**
  - Minimization of Rayleigh quotient
  - Fixed subspace size
  - Conjugate gradient algorithm
- **Newton-based schemes:**
  - Block Davidson
  - Block Jacobi-Davidson

Important note: **We use a preconditioner N for K ($\sigma = 0$).**

Sandia National Laboratories

Block DACG

- The algorithm minimizes the Rayleigh quotient on the space

$$\mathrm{Span}\{X^{(i)}, P^{(i)}\}$$

- The search directions $P^{(i)}$ are defined by

$$P^{(i)} = -N^{-1}(KX^{(i)} - MX^{(i)}\Lambda^{(i)}) + P^{(i-1)}B^{(i)}$$

- The matrix $B^{(i)}$ uses the Bradbury-Fletcher formula

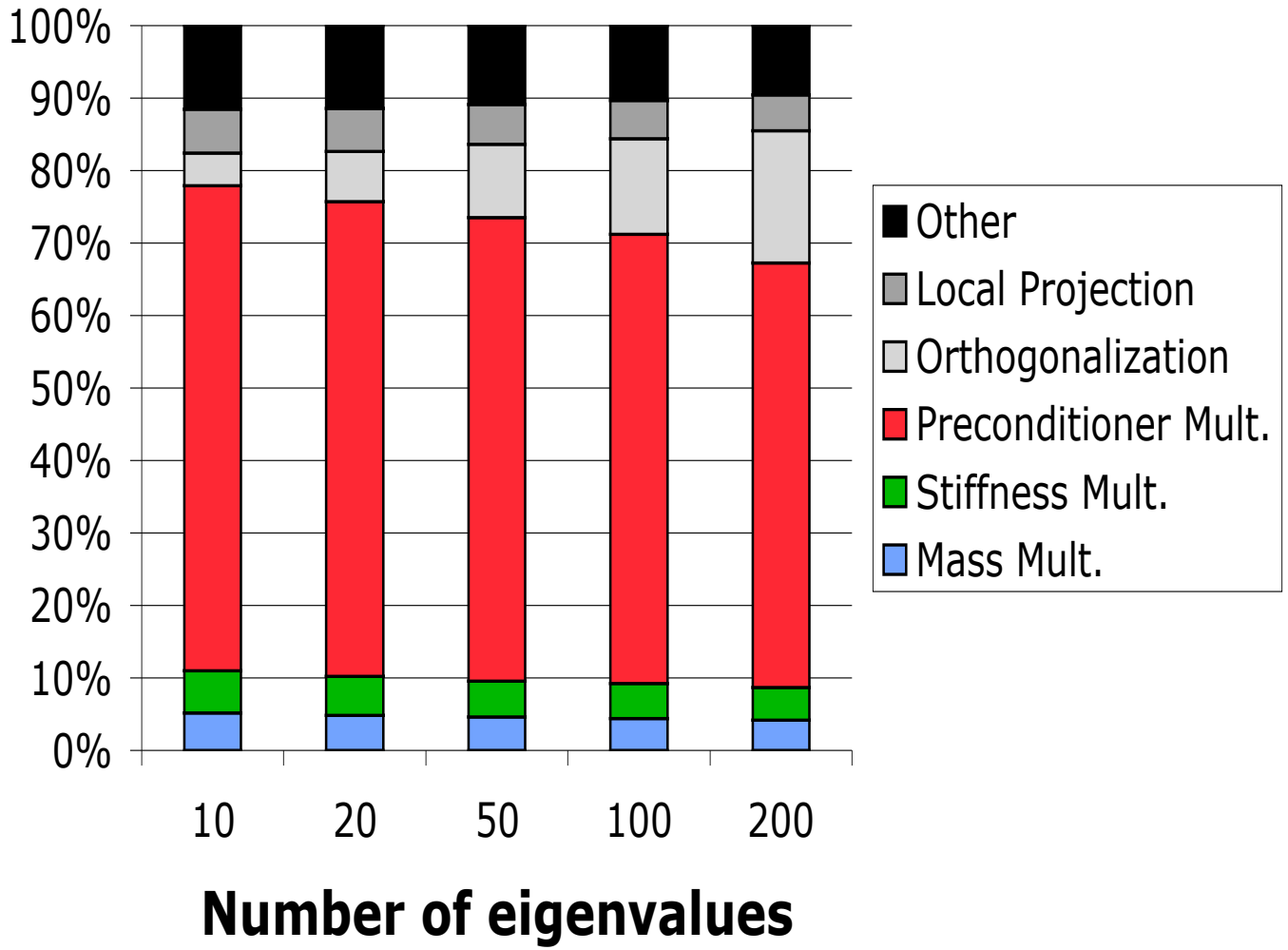
$$B^{(i)} = \frac{\mathrm{diag}\left[\left(KX^{(i)} - MX^{(i)}\Lambda^{(i)}\right)^{T} N^{-1}\left(KX^{(i)} - MX^{(i)}\Lambda^{(i)}\right)\right]}{\mathrm{diag}\left[\left(KX^{(i-1)} - MX^{(i-1)}\Lambda^{(i-1)}\right)^{T} N^{-1}\left(KX^{(i-1)} - MX^{(i-1)}\Lambda^{(i-1)}\right)\right]}$$

- The classical Gram-Schmidt algorithm is used for orthogonalization.
- The eigenvectors are deflated at convergence.

Reference: L. Bergamaschi and M. Putti (2002)

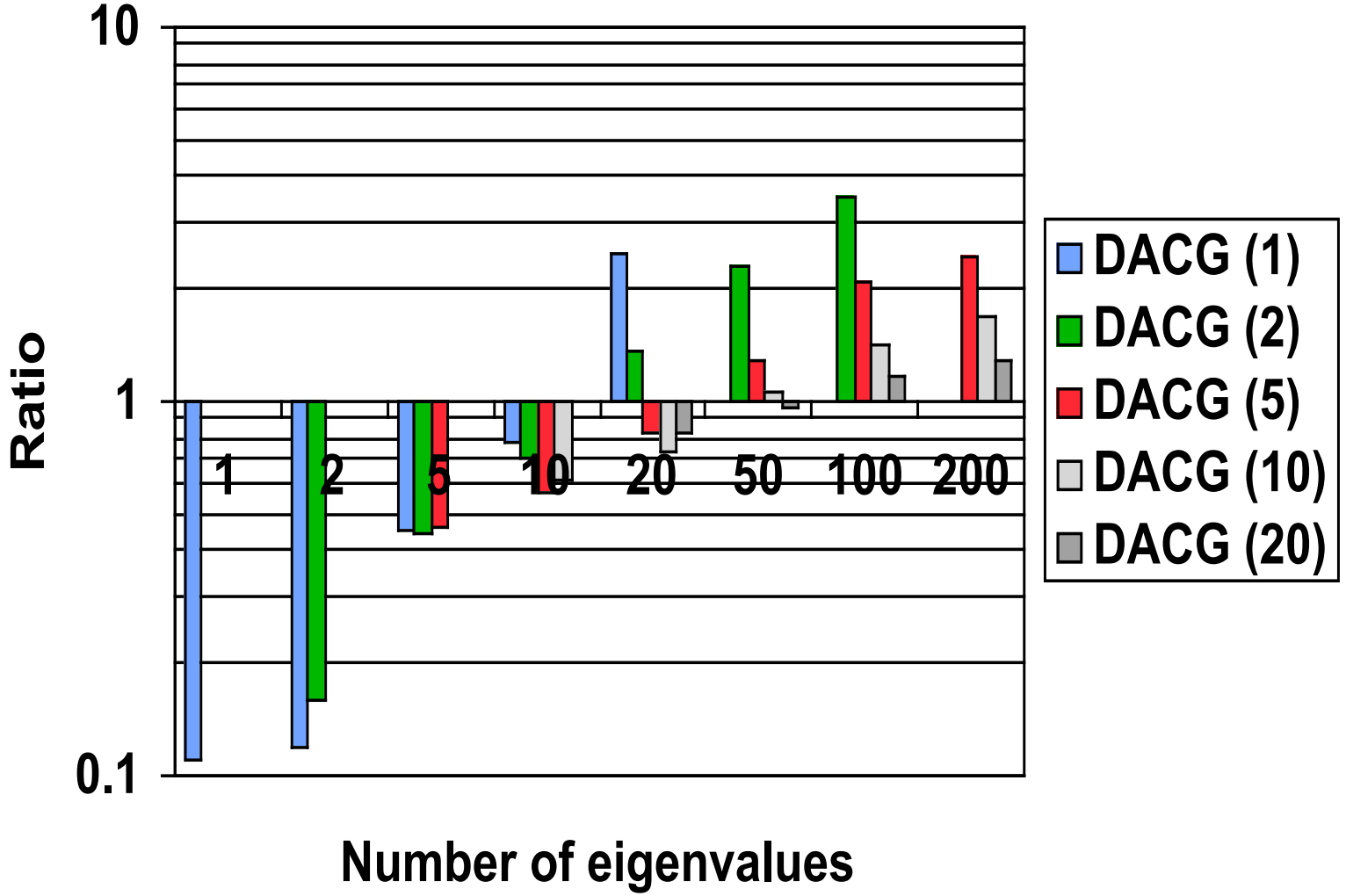
Sandia National Laboratories

Time Distribution for DACG (10)



Sandia National Laboratories

Relative Time vs ARPACK



Sandia National Laboratories

LOBPCG

- The incremental idea is to use a three-term recurrence à la Lanczos.
- The algorithm minimizes the Rayleigh quotient on the space

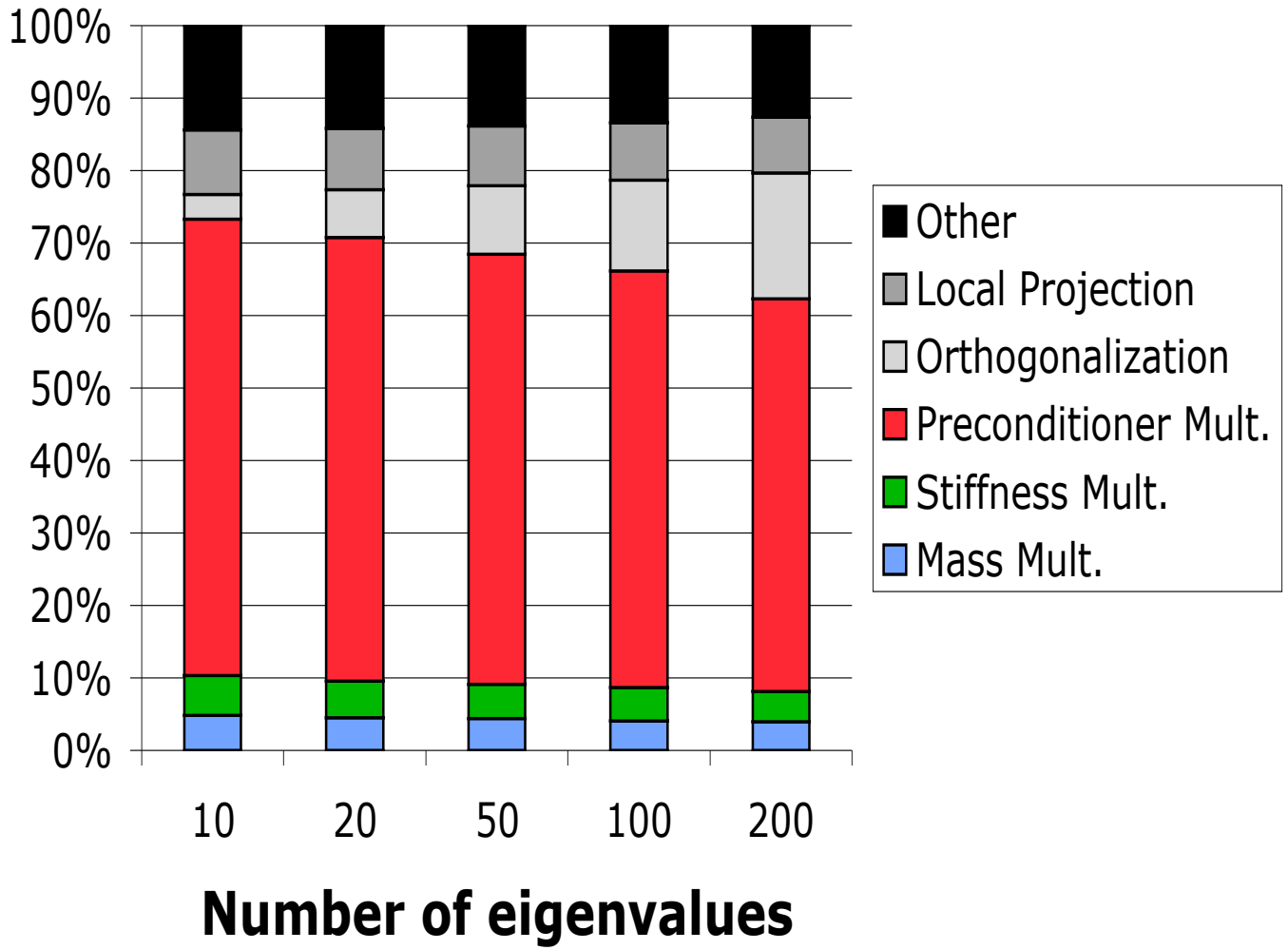
$$\mathrm{Span}\{X^{(i)}, X^{(i-1)}, N^{-1}(KX^{(i)}-MX^{(i)}\Lambda^{(i)})\}$$

- The classical Gram-Schmidt algorithm is used for orthogonalization.
- The eigenvectors are deflated at convergence.

Reference: A. Knyazev (2001)

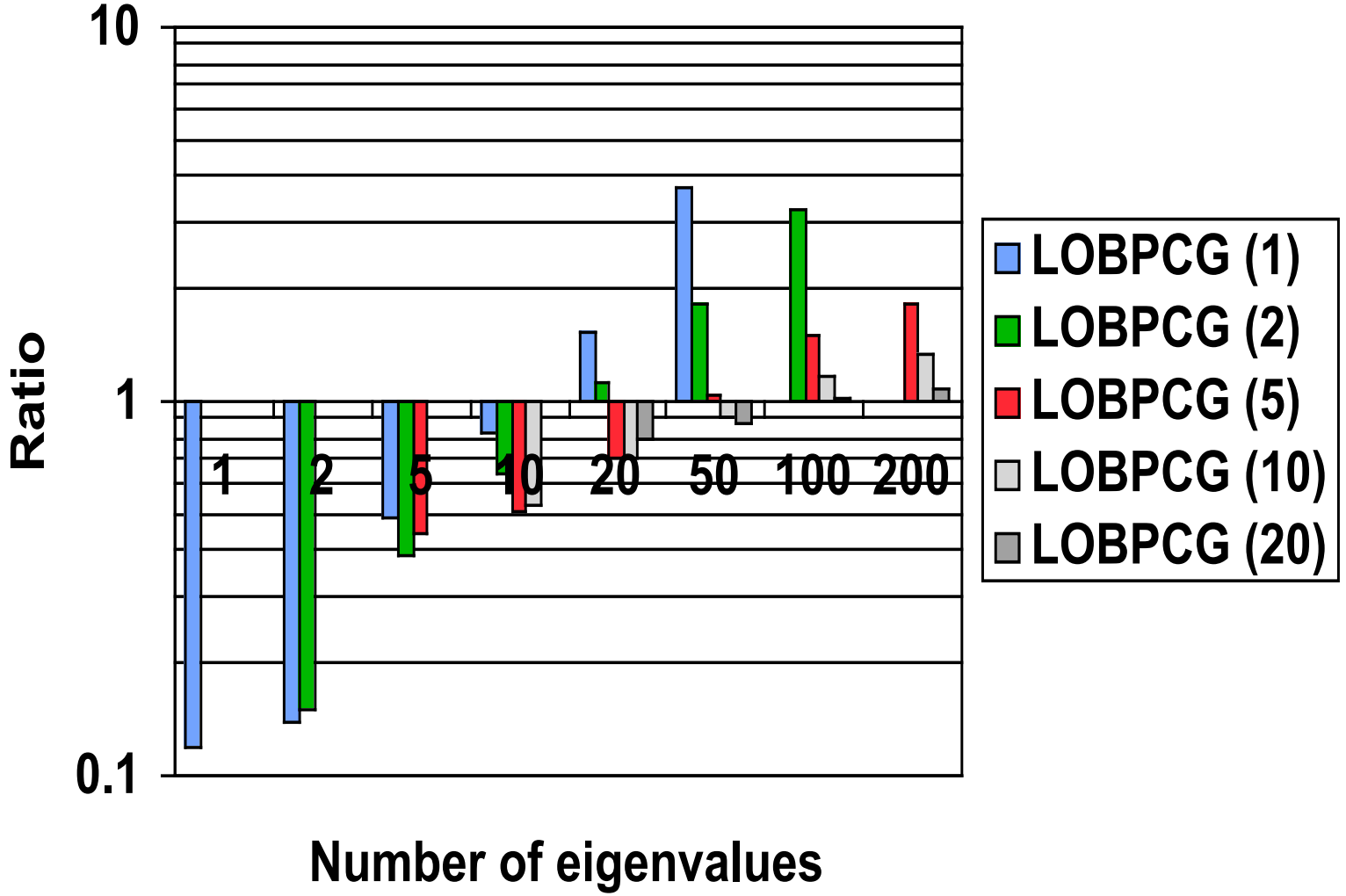
Sandia National Laboratories

Time Distribution for LOBPCG (10)



Sandia National Laboratories

Relative Time vs ARPACK



Sandia National Laboratories

Davidson Algorithm

- The algorithm minimizes the Rayleigh quotient on the space

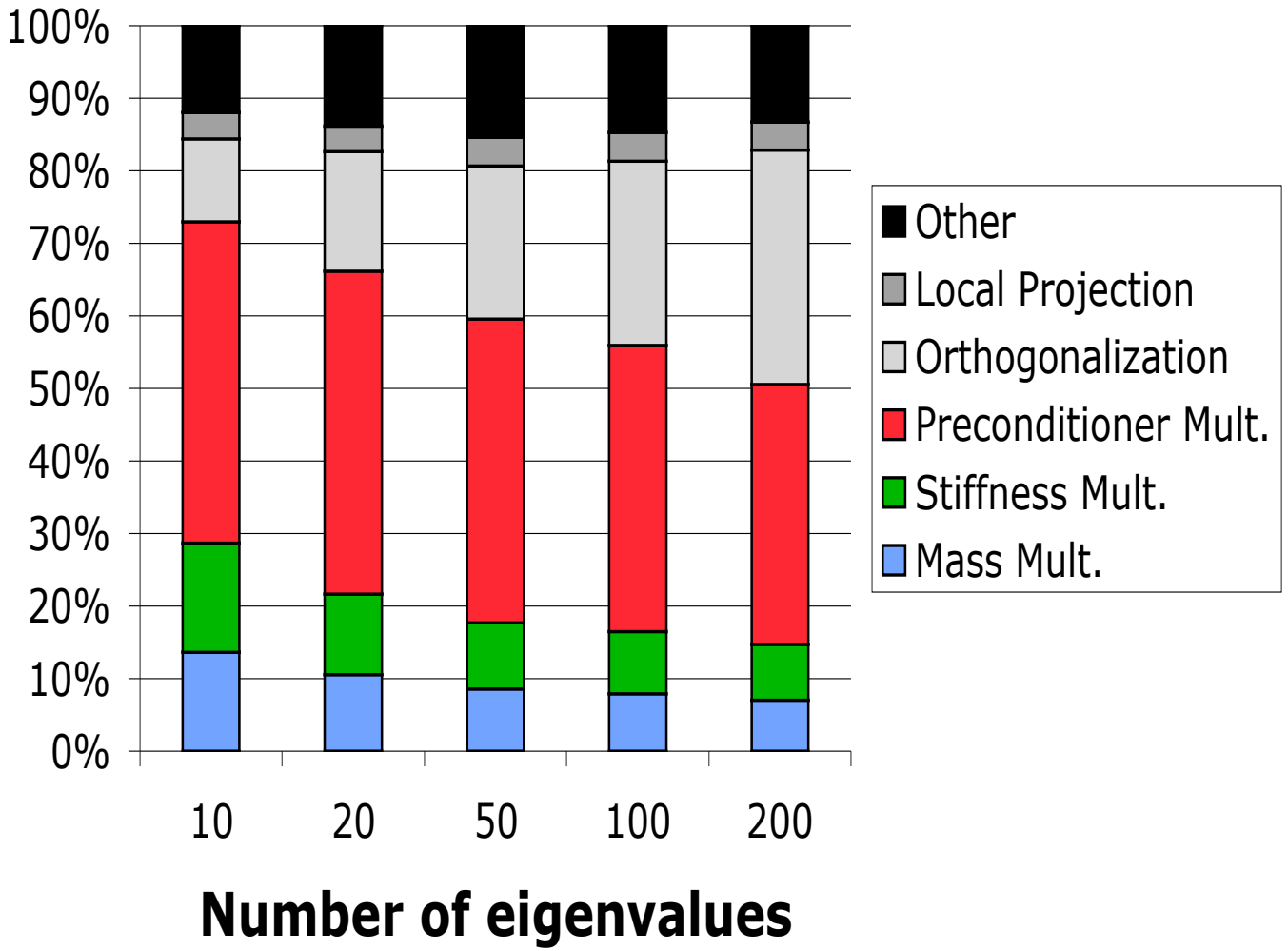
$$\text{Span}\{X^{(0)},\ N^{-1}(KX^{(0)}-MX^{(0)}\Lambda^{(0)}),\ \ldots$$

$$N^{-1}(KX^{(m-1)}-MX^{(m-1)}\Lambda^{(m-1)}),\ N^{-1}(KX^{(m)}-MX^{(m)}\Lambda^{(m)})\}$$

- For nev eigenvalues requested, the subspace is restarted when the size reaches 2*nev.
- The preconditioner N is fixed for all the computation.
- An M-orthonormal basis is generated for the subspace.
- The eigenvectors are deflated at convergence.

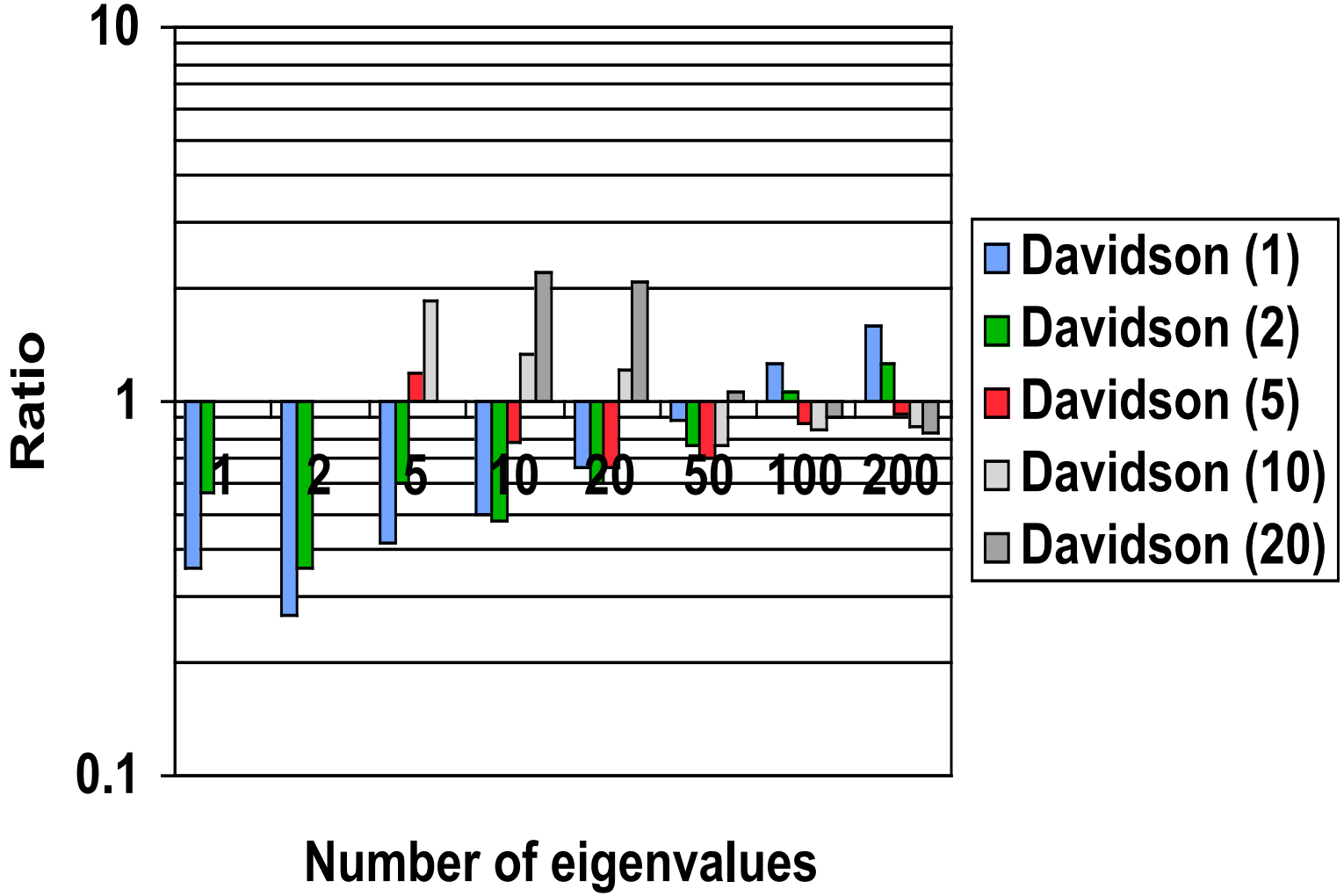
Sandia National Laboratories

Time Distribution for Davidson (10)



Sandia National Laboratories

Relative Time vs ARPACK



Sandia National Laboratories

Jacobi Davidson

- The algorithm is based on the Davidson algorithm.
- The correction equation improves the computed eigenvector

$$(I-MQQ^T)(K-\lambda M)(I-QQ^TM)Z = -(I-MQQ^T)(KX-MX\Lambda)$$

- The correction equation is solved with preconditioned conjugate gradient

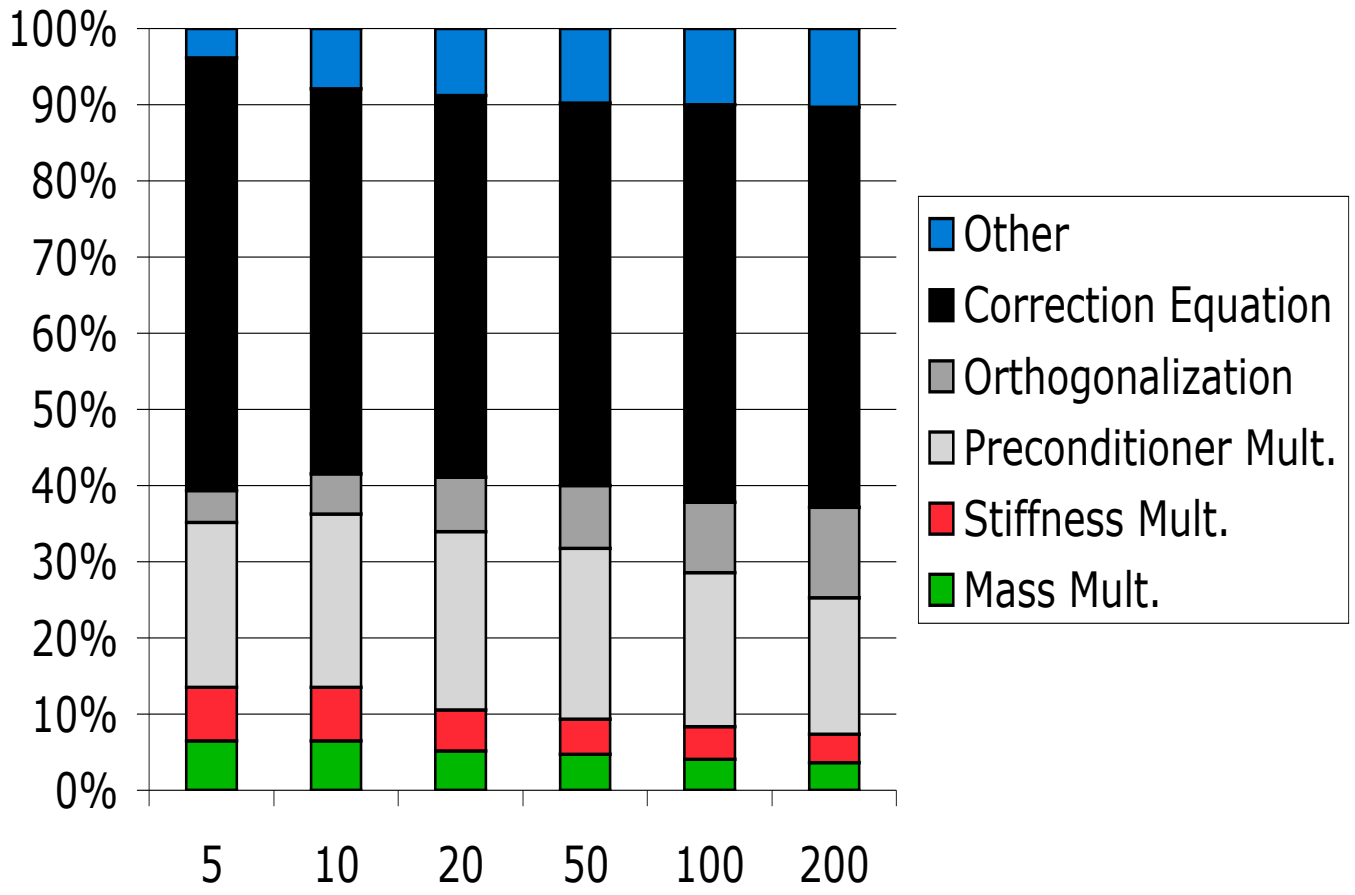
$$(I-MQQ^T)N(I-QQ^TM)$$

- The preconditioner N is fixed for all the computation.
- An M-orthonormal basis is generated for the subspace.
- The eigenvectors are deflated at convergence.

Reference: G. Sleijpen and H. Van Der Vorst (1996), Y. Notay (2001)

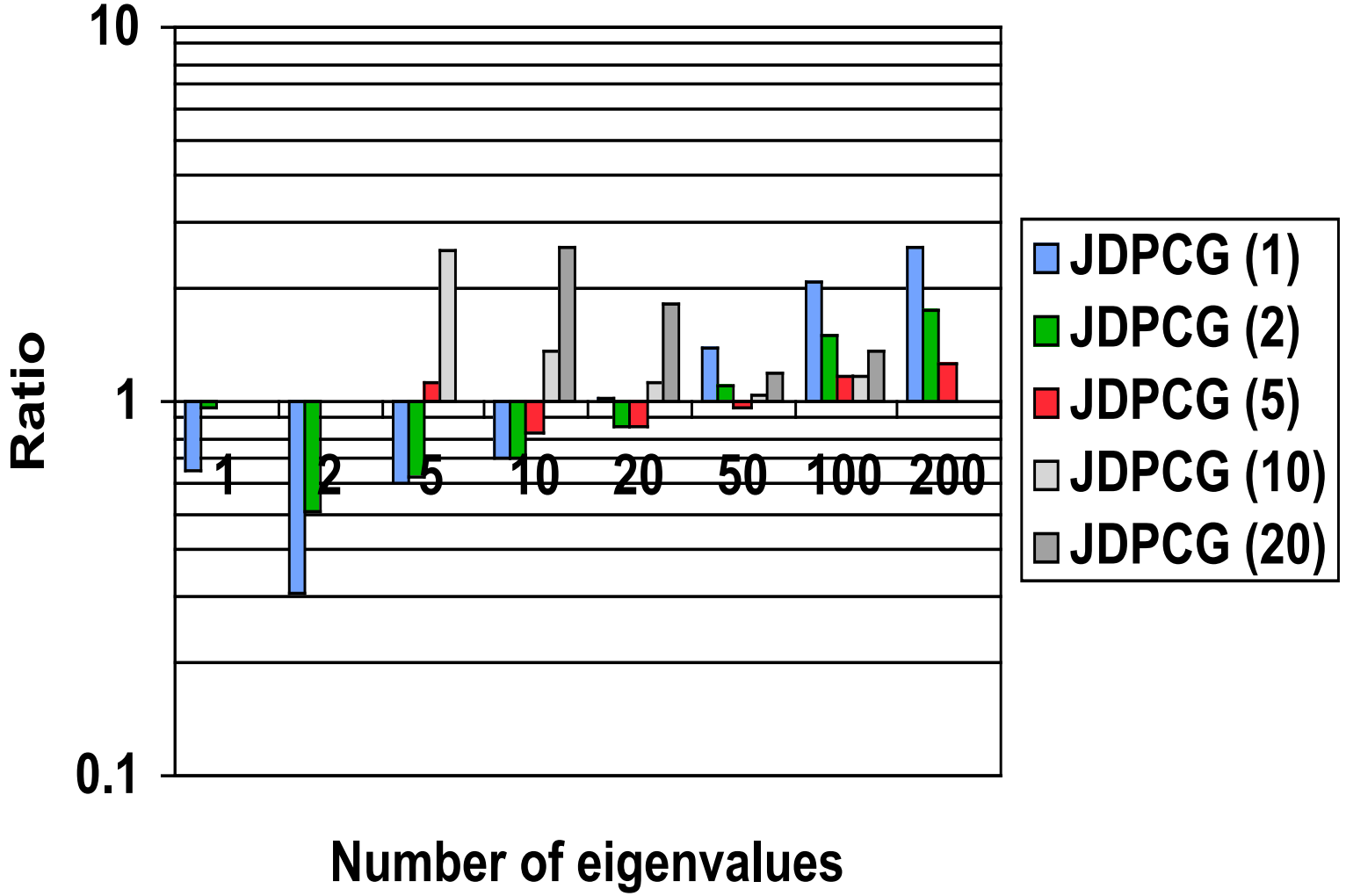
Sandia National Laboratories

Time Distribution for JDPCG (5)



Sandia National Laboratories

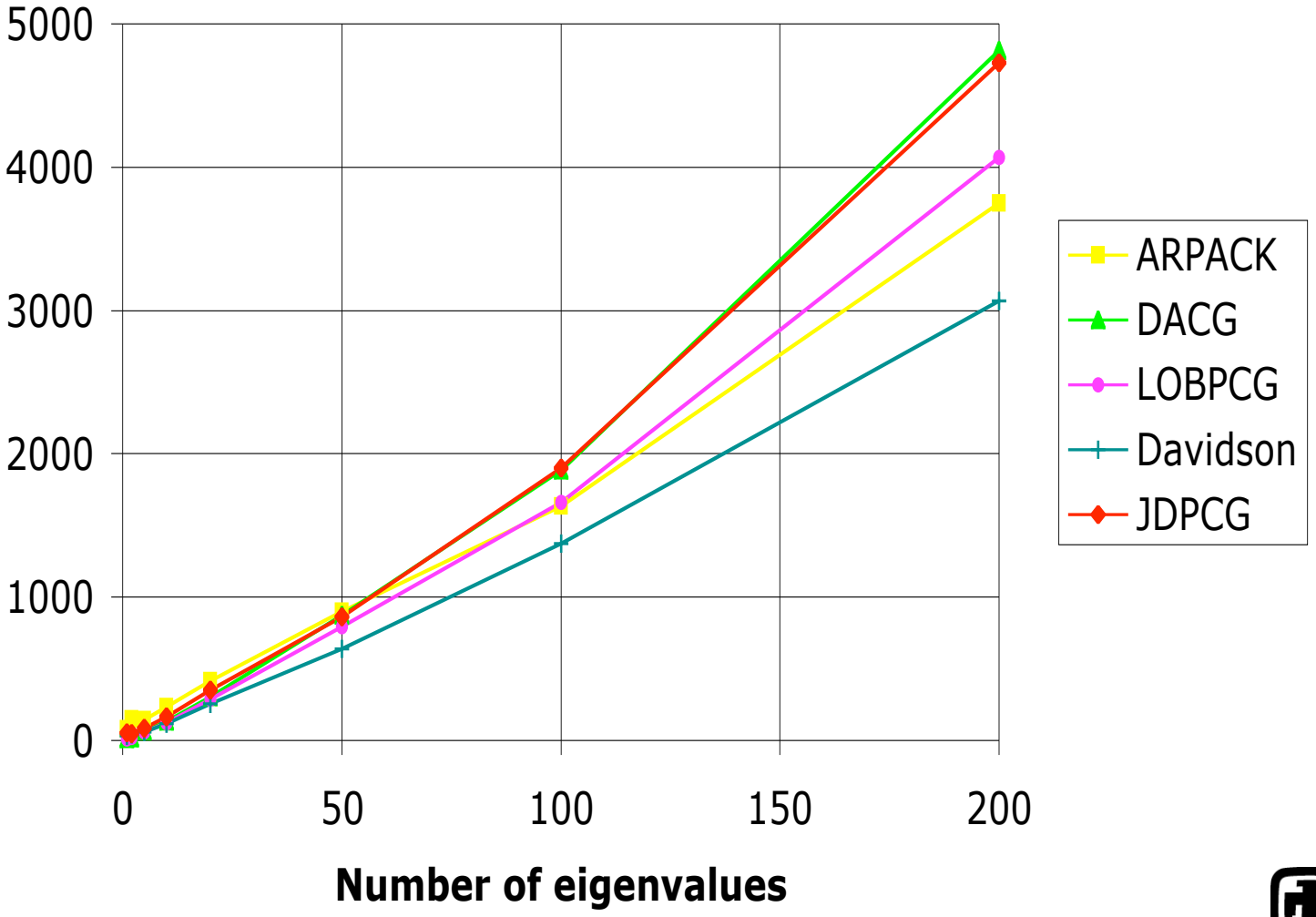
Relative Time vs ARPACK



Sandia National Laboratories

Time Summary

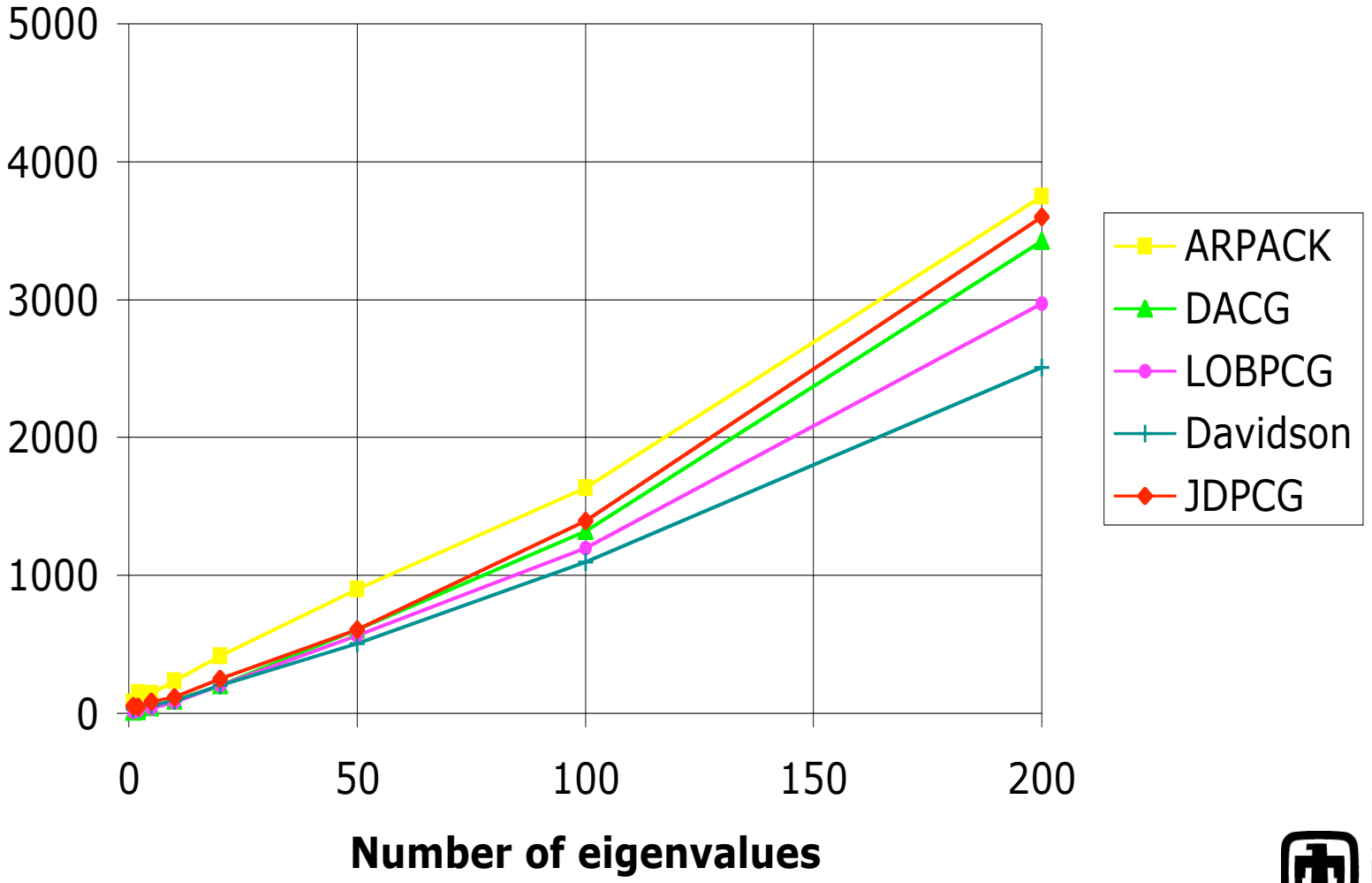
- **Preconditioner applications are not blocked.**



Sandia National Laboratories

Time with Blocked Preconditioner

- **Speedup of 2 for preconditioner-vector applications**



Sandia National Laboratories

Memory Cost

Lanczos method:

- ncv (= 2*nev) vectors
- 3 vectors
- Arrays for $O(nev^2)$

Block DACG:

- nev vectors
- 8 blocks of vectors
- Arrays for $O(b^2)$

LOBPCG:

- nev vectors
- 10 blocks of vectors
- Arrays for $O(b^2)$

Davidson:

- ncv (= 2*nev) vectors
- 4 blocks of vectors
- Arrays for $O(nev^2)$

Jacobi-Davidson:

- ncv (= 2*nev) vectors
- 2*nev vectors
- 5 blocks of vectors
- Arrays for $O(nev^2)$

Size of block = b

Summary

Comments:

- Applications of preconditioner should be blocked.
- For this problem, Davidson is a good simple algorithm.
- For this problem, Jacobi-Davidson did not improve the performance.
- Ultimately, orthogonalization becomes the bottleneck.

Future work:

- Test the solvers on elasticity problems
- Evaluate the CMS method

References

• J. Bennighof, M. Kaplan, and M. Muller, *"Extending the frequency response capabilities of automated multilevel substructuring",* no. AIAA-2000-1574, April 2000.

• L. Bergamaschi and M. Putti, *"Numerical comparison of iterative eigensolvers for large sparse symmetric positive definite matrices",* CMAME, v. 191, p. 5233-5247, 2002.

• M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson and D. Rixen, *"Application of the FETI Method to ASCI Problems: Scalability Results on One-Thousand Processors and Discussion of Highly Heterogeneous Problems",* IJNME, v. 47, p. 513-536, 2000.

• M. Bhardwaj, K. Pierson, G. Reese, T. Walsh, D. Day, K. Alvin, J. Peery, C. Farhat and M. Lesoinne, *"Salinas: A Scalable Software for High-Performance Structural and Solid Mechanics Simulations",* Proceedings of the IEEE/ACM SC2002 Conference, Baltimore, Maryland, 2002.

• A. Kropp and D. Heiserer, *"Efficient broadband vibro-acoustic analysis of passenger car bodies using an FE-based component mode synthesis approach",* WCCM V, 2002.

• A. Knyazev, *"Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method"*, SIAM J. Sci. Comput., v. 23, p. 517-541, 2001.

• R. Lehoucq, D. Sorensen, and C. Yang, *"ARPACK users' guide: Solution of large-scale eigenvalue problems by implicitly restarted Arnoldi methods"*, SIAM, Philadelphia, PA, 1998.

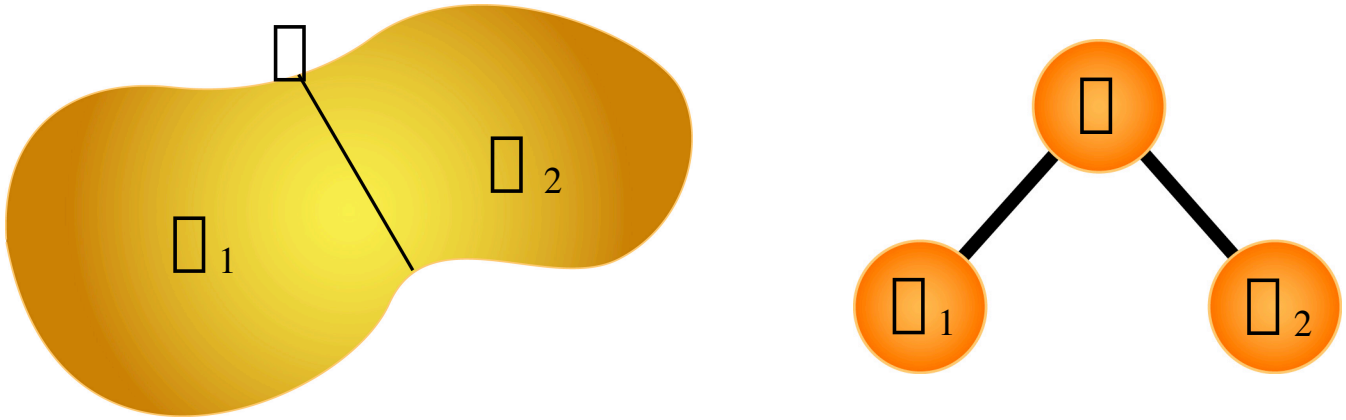
• Y. Notay, *"Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem",* Numer. Linear Algebra Appl., v. 9, p. 21-44, 2001.

• G. Sleijpen and H. Van Der Vorst, *"A Jacobi-Davidson method for eigenvalue problems"*, SIAM J. Matrix Anal. Appl., v. 17, p. 401-425, 1996.

Sandia National Laboratories

Component Mode Synthesis (CMS)

Idea: **Compute the first eigenmodes of a structure that can be subdivided into substructures on each of which the first eigenmodes are known.**



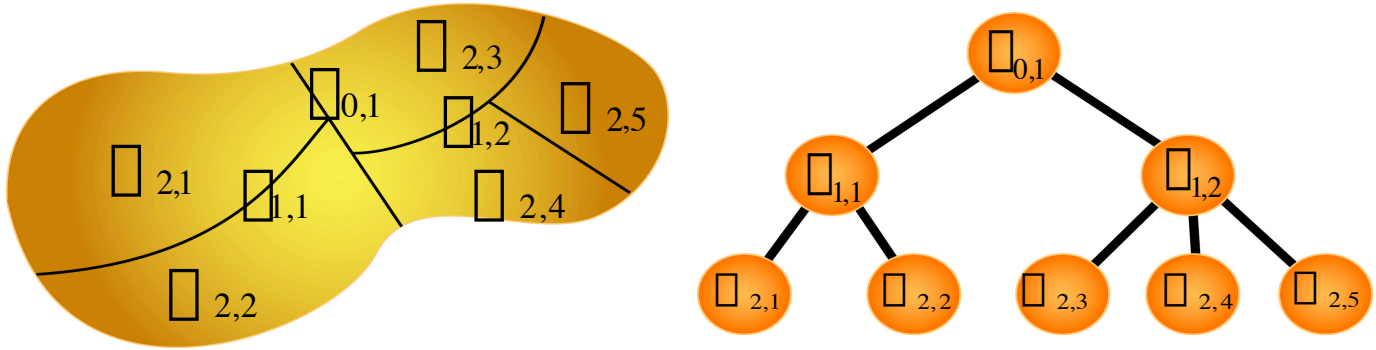
1. **Compute a few local eigenmodes in Ω_1**
2. **Compute a few local eigenmodes in Ω_2**
3. **Compute a few local eigenmodes on Γ**
4. **Approximate the global eigenmodes with the local modes**

Reference: W. Hurty (1965), R. Craig and M. Bampton (1968), F. Bourquin (1991)

Sandia National Laboratories

Component Mode Synthesis (CMS)

- **Automated MultiLevel Substructuring method (AMLS)**
  - **Apply the previous idea recursively**
  - **Use multilevel nested dissection to partition the domain**



Loop over the substructures

…

Solve a local generalized eigenproblem

…

…

Solve a coarse eigenproblem for synthesis

Sandia National Laboratories

Component Mode Synthesis (CMS)

- **Automated MultiLevel Substructuring method (AMLS)**

  - **For shells and plates (2D-like) problems, AMLS brings a major improvement in CPU time and memory costs over classical methods.**

    - ***Works of J. Bennighof and his group at UT Austin.***
    - ***Experiments of A. Kropp and D. Heiserer (BMW).***

  - **Very few 3D computations for real problems have been done !**

    - ***Work in progress (issues similar for a multifrontal solver)***