
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrived as an official Department of the Army position, policy or decision, unless so designated by other documentation.

This grant enabled the purchase of an Intel multiprocessor consisting of eight multicore nodes interconnected via an infiniband. Each node contains 24 cores. This parallel computing platform has been used by my research group in the early stages of developing large sparse linear system and symmetric eigenvalue problem solvers (ARO grant W911NF-07-R-0003-04) that are suitable for parallel architectures containing hundreds of multicore nodes (thousands of cores). Once our parallel solvers obtain the correct solutions, and perform properly on this 8-node platform, the codes are ported to much larger parallel computing platforms at the Intel Corporation. Two Ph.D.

large sparse linear system solvers, large sparse symmetric eigenvalue problem solvers, parallel programming
Report Title

ABSTRACT
This grant enabled the purchase of an Intel multiprocessor consisting of eight multicore nodes interconnected via an infiniband. Each node contains 24 cores. This parallel computing platform has been used by my research group in the early stages of developing large sparse linear system and symmetric eigenvalue problem solvers (ARO grant W911NF-07-R-0003-04) that are suitable for parallel architectures containing hundreds of multicore nodes (thousands of cores). Once our parallel solvers obtain the correct solutions, and perform properly on this 8-node platform, the codes are ported to much larger parallel computing platforms at the Intel Corporation. Two Ph.D. dissertations have resulted from the above grant (awarded by Dr. Myers), and this DURIP grant (awarded by Dr. Coyle).

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received Paper

TOTAL:

Number of Papers published in non peer-reviewed journals:

(c) Presentations


Number of Presentations: 3.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received  Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received  Paper

TOTAL:

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received  Paper

01/05/2016  2.00  Yao Zhu, Ahmed H. Sameh. SPIKE+: A family of parallel hybrid sparse linear system solvers, Journal of Computational and Applied Mathematics (05 2015)

TOTAL:  1
Books

Received  Book

TOTAL:

Received  Book Chapter

01/05/2016  1.00 Yao Zhu, Ahmed H. Sameh. How to generate effective block Jacobi preconditioners for solving large sparse linear systems, Yet to be published: Birkhauser, (02 2016)

TOTAL:  1

Patents Submitted

Patents Awarded

Awards

Graduate Students

<table>
<thead>
<tr>
<th>NAME</th>
<th>PERCENT_SUPPORTED</th>
<th>Discipline</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yao Zhu</td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>Alicia Klinvex</td>
<td>0.10</td>
<td></td>
</tr>
</tbody>
</table>

FTE Equivalent: 0.60

Total Number: 2
Names of Post Doctorates

<table>
<thead>
<tr>
<th>NAME</th>
<th>PERCENT_SUPPORTED</th>
<th>FTE Equivalent:</th>
<th>Total Number:</th>
</tr>
</thead>
</table>

Names of Faculty Supported

<table>
<thead>
<tr>
<th>NAME</th>
<th>PERCENT_SUPPORTED</th>
<th>National Academy Member</th>
<th>FTE Equivalent:</th>
<th>Total Number:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahmed H. Sameh</td>
<td>0.10</td>
<td></td>
<td>0.10</td>
<td>1</td>
</tr>
</tbody>
</table>

Names of Under Graduate students supported

<table>
<thead>
<tr>
<th>NAME</th>
<th>PERCENT_SUPPORTED</th>
<th>FTE Equivalent:</th>
<th>Total Number:</th>
</tr>
</thead>
</table>

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 0.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields: 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense: 0.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields: 0.00

Names of Personnel receiving masters degrees

<table>
<thead>
<tr>
<th>NAME</th>
<th>Total Number:</th>
</tr>
</thead>
</table>

Names of personnel receiving PHDs

<table>
<thead>
<tr>
<th>NAME</th>
<th>Total Number:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yao Zhu</td>
<td>2</td>
</tr>
<tr>
<td>Alicia Klinvex</td>
<td></td>
</tr>
</tbody>
</table>
Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

Two classes of parallel solvers have been developed. The first is a family of parallel sparse linear system solvers -- the PSPIKE family (Spike/Pardiso hybrid solvers). The second is a family of Trace Minimization eigensolvers for the symmetric standard and generalized eigenvalue problems. Both classes of solvers proved to be competitive with existing solvers in being more robust and more scalable on parallel architectures with superior speeds. In particular the TraceMin eigensolvers developed through these two ARO grants have been adopted by the DOE Trilinos project at Sandia National Laboratories.

Technology Transfer
TraceMIN: a scalable parallel algorithm for large sparse symmetric eigenvalue problems

Alicia Klinvex
Sandia National Labs.

Ahmed H. Sameh
Computer Science, Purdue University

Support: ARO, Intel, NSF
St Girons: June 27 – July 2; 2015
Symmetric Generalized Eigenvalue Problem

\[ Ax = \lambda Bx \]

A is sparse and symmetric
B is sparse and s.p.d.

Obtain the \( p \) eigenvalues closest to 0 and their corresponding eigenvectors, \( p \ll n \)
The Trace minimization scheme

\[ \min_{Y^TBY = I_p} \text{tr} (Y^TAY) = \sum_{i=1}^{p} \lambda_i \]

\[ \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_p < \lambda_{p+1} \leq \cdots \leq \lambda_n \]

\[ Y \in \mathbb{R}^{n \times p} \quad ; \quad p \ll n. \]

A.S. & J. Wisniewski: SINUM, 1982
Extensions of this optimization problem for definite matrix pair

• B: indefinite nonsingular
• B: indefinite singular
  — Liang, Li, and Z. Bai

A and B are called “definite pair” if there exists a scalar $\mu$ such that $(A - \mu B)$ is s.p.d, or s.p.s.d
solve

\[
\begin{pmatrix}
A & BY_k \\
Y_k^T B & O
\end{pmatrix}
\begin{pmatrix}
\Delta_k \\
L_k
\end{pmatrix}
= 
\begin{pmatrix}
AY_k \\
O
\end{pmatrix} \quad (*)
\]

or

\[
\begin{pmatrix}
A & BY_k \\
Y_k^T B & O
\end{pmatrix}
\begin{pmatrix}
Y_k - \Delta_k \\
-L_k
\end{pmatrix}
= 
\begin{pmatrix}
O \\
I_p
\end{pmatrix} \quad (**) 
\]

• \((Y_k - \Delta_k)\) is used to generate \(Y_{k+1}\)
• \(\text{tr } (Y_{k+1} A Y_{k+1}) \leq \text{tr}(Y_k A Y_k)\)
• Different schemes (direct/iterative) for solving (*) or (**) 
• **TraceMIN does not require low relative residuals in solving (*) or (**)**
a) Using a direct sparse linear system solver (Pardiso)

- **Solve**
  \[ AZ_k = BY_k \quad \text{for } Z_k \]

- **Solve**
  \[
  \begin{pmatrix}
  I & Z_k \\
  Y_k^T B & O
  \end{pmatrix}
  \begin{pmatrix}
  Y_k - \Delta_k \\
  -L
  \end{pmatrix}
  =
  \begin{pmatrix}
  O \\
  I_p
  \end{pmatrix}
  \]

  for \( Y_k - \Delta_k \)
b) Use a Krylov subspace method to solve

\[(I - P_k)A(I - P_k)\Delta_k = (I - P_k)AY_k\]

for \(\Delta_k\)

where

\[P_k = BY_k(Y_k^TB^2Y_k)^{-1}Y_k^TB.\]

e.g. CG, or Minres; rel. res \(\leq 10^{-6}\)
c) Solve

\[
\begin{pmatrix}
A & BY \\
Y^T B & O
\end{pmatrix}
\begin{pmatrix}
Y - \Delta \\
- L
\end{pmatrix} =
\begin{pmatrix}
O \\
I_p
\end{pmatrix}
\]

via a Krylov subspace method with the preconditioner

\[
M = \text{diag} \left( \hat{A}, Y^T B \hat{A}^{-1} B Y \right)
\]

\(\hat{A}\) is an approximation of \(A\)

*Murphy, Golub, Wathen: SISC 1999*
TraceMIN algorithm

• Choose initial \((n \times s)\) block \(V;\) \(s \approx 2p\)
• Repeat until convergence
  – 1. \(B\)-orthonormalize \(V\)
  – 2. Form approximate eigenpairs
    \[
    Y^T A Y = \Sigma, \quad Y^T B Y = I; \quad \Sigma = \text{diag} (\sigma_1, \sigma_2, \ldots, \sigma_s)
    \]
  – 3. \(R = AY - BY\Sigma,\) check \(\text{norm}(r_k) / \sigma_k \leq 10^{-5}\)
  – 4. Solve saddle-point problem for \(\Delta\) or \(Y - \Delta\)
  – 5. update \(V = Y - \Delta\)
Convergence

• Convergence bounded by $| \lambda_k / \lambda_{s+1} |$

• Origin shifts can lead to faster convergence

  $-(A - \mu_k B) x_k = (\lambda_k - \mu_k) x_k$

  $- |(\lambda_k - \mu_k) / \lambda_{s+1}|

• If $A$ is s.p.d, then

  $\Phi_k = (y_k - x_k)^T A (y_k - x_k)$

  is reduced asymptotically by a factor of

  $(\lambda_k / \lambda_{s+1})^2$
Simultaneous Iterations & TraceMIN

• Simultaneous Iterations is mathematically equivalent to TraceMIN.
• In each step of Simultaneous Iterations, we need to solve systems of the form $A V = B Y$.
• If we solve such systems, as well as TraceMIN saddle-point problems using Minres, for example, with a modest stopping criterion, i.e. rel. res. $\leq 0.5$, we get vastly different behavior in reduction of the trace.
$A x = \lambda B x$

$n = 100; \ p = 4; \ s = 8$

$\kappa_2 (A) \sim \kappa_2 (B) = 1000$

Error in trace < $10^{-8}$

< 25 iterations

Minres: rel. res $\leq 0.5$
### Three trace minimization schemes for the symmetric eigenvalue problem

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subspace dimension</strong></td>
<td>constant</td>
<td>expanding</td>
<td>expanding</td>
</tr>
<tr>
<td><strong>Choice of shifts</strong></td>
<td>conservative</td>
<td>aggressive</td>
<td>conservative</td>
</tr>
</tbody>
</table>

**How to handle expanding subspaces**

V := [V, Δ]

*Upon restart, retain only relevant part of the subspace*
Performance Results: a sample

Codes in Trilinos framework

TraceMIN & TraceMIN-Davidson (TD) vs:

1. The DOE Trilinos Anasazi package –
   - Krylov Schur (KS) [G.W. Stewart]; (BKS) [Saad]
   - Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG), [Knyazev]
   - Riemannian Trust Region (RTR) ** [Baker et al]

2. The SLEPc package -- Jacobi-Davidson (JD) **

** TraceMIN-based
Robustness

• Obtain the 4 smallest eigenpairs of the standard eigenvalue problem: $A \mathbf{x} = \lambda \mathbf{x}$

• Source: Tim Davis sparse matrix collection

• 3 benchmarks for which $A$ is symmetric positive definite

• 1 benchmark for which $A$ is indefinite

• ✔: success

• ✗: failure (no convergence or wrong multiplicity)
Computing Platform

• Intel cluster: Endeavor
• Infiniband interconnection network
• Each node consists of two 14-core Intel processors running at 2.6 GHz
• 64 GB per node
• Hybrid programming paradigm MPI/OpenMP
## Robustness

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Type (n)</th>
<th>TD</th>
<th>BKS</th>
<th>LOBPCG</th>
<th>RTR</th>
<th>KS</th>
<th>JD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>spd 64 M</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Flan_1565</td>
<td>spd 1.5 M</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>Hook_1498</td>
<td>spd 1.5 M</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>nlpkkt240</td>
<td>Indef. 28 M</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
</tr>
</tbody>
</table>
Time(algorithm $X$) / Time(TD)

- **32 nodes (24 cores per node)**

<table>
<thead>
<tr>
<th>matrix</th>
<th>Type</th>
<th>$n$</th>
<th>TD</th>
<th>BKS</th>
<th>LOBPCG</th>
<th>RTR</th>
<th>KS</th>
<th>JD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hook_1498</td>
<td>spd</td>
<td>1.5 M</td>
<td>1.0</td>
<td>✓</td>
<td>✓</td>
<td>3.3</td>
<td>✓</td>
<td>1.5</td>
</tr>
</tbody>
</table>

- **128 nodes (24 cores per node)**

<table>
<thead>
<tr>
<th>matrix</th>
<th>Type</th>
<th>$n$</th>
<th>TD</th>
<th>BKS</th>
<th>LOBPCG</th>
<th>RTR</th>
<th>KS</th>
<th>JD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>spd</td>
<td>64 M</td>
<td>1.0</td>
<td>19.7</td>
<td>3</td>
<td>1.6</td>
<td>✓</td>
<td>--</td>
</tr>
<tr>
<td>Flan_1565</td>
<td>spd</td>
<td>1.5 M</td>
<td>1.0</td>
<td>✓</td>
<td>✓</td>
<td>1.3</td>
<td>✓</td>
<td>2.1</td>
</tr>
<tr>
<td>nlpkkt240</td>
<td>indef</td>
<td>28 M</td>
<td>1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>6.6</td>
</tr>
</tbody>
</table>
computing the 4 smallest eigenpairs of nlpkkt240
**TraceMIN**: multisectioning & sampling  
*(codes in Fortran 90)*

- **Multisectioning:**
  - For obtaining a large number of interior eigenvalues and their corresponding eigenvectors

- **Sampling:**
  - Obtaining several eigenpairs in the neighborhood of a given number of points within the spectrum
Multisectioning

• Using several shifts $\alpha_j$ we can obtain that interval $(0, \tau)$ that contains the number of eigenpairs desired.

• Divide the interval under consideration into many subintervals, one subinterval per node; each containing roughly the same number of eigenvalues (e.g. 20), if possible

• For each node $j$, we use our shared-memory TraceMIN eigensolver to obtain the smallest eigenpairs of $(A - \alpha_j B) x = (\lambda - \alpha_j) B x$. 

With the shifts $\alpha$, we get saddle-point problems of the form:

\[
\begin{pmatrix}
A - \alpha B & BY_k \\
Y_k^T B & O
\end{pmatrix}
\begin{pmatrix}
Y_k - \Delta_k \\
-L_k
\end{pmatrix} = \begin{pmatrix} O \\ I_p \end{pmatrix}.
\]

A sparse direct solver (Pardiso) is used to Obtain the factorization of the indefinite (1,1) block:

\[
A - \alpha B = LDL^T.
\]
Multisectioning – an illustration

• Obtain the eigenpairs corresponding to $[\alpha, \beta]$
• $[\alpha, \beta]$ is divided into 4 sub-intervals
• Use five nodes
• Except for Node #1, each node performs 2 factorizations:
  – one for computing the inertia (blue), and
  – one for the TraceMIN iteration (red).
• Node 1 performs one factorization with shift $\alpha$
A NASTRAN benchmark

Car body dynamics —

- $A x = \lambda B x$ (order: 1.5 M, and 7.2 M)
- $A = A^T$, $B := s.p.d.$
- $\text{norm}(R,\text{inf}) \leq 10^{-6}$
- Compute the smallest 1000 eigenpairs using 129 nodes.
Sparsity Structure of $A$
$n \sim 7.2 \, M$
TraceMIN w/ multisectioning

vs. Intel’s Math Kernel Library’s FEAST

<table>
<thead>
<tr>
<th>Problem</th>
<th>size</th>
<th># of smallest $\lambda$s</th>
<th>TraceMIN (seconds)</th>
<th>FEAST (seconds)</th>
<th>Speed Improve.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nastran 1</td>
<td>1.5 M</td>
<td>1000</td>
<td>59</td>
<td>121</td>
<td>2.2</td>
</tr>
<tr>
<td>Nastran 2</td>
<td>7.2 M</td>
<td>1000</td>
<td>418</td>
<td>Failure</td>
<td></td>
</tr>
</tbody>
</table>

All Trilinos eigensolvers failed on this NASTRAN benchmark
**TraceMIN w/ multisectioning**

**vs. Intel’s Math Kernel Library’s FEAST**

*using 129 nodes; norm \((r_k, \inf)/\sigma_k \leq 10^{-5}\)*

<table>
<thead>
<tr>
<th>Problem</th>
<th>size</th>
<th># of (\lambda)s interval</th>
<th>TraceMIN (seconds)</th>
<th>FEAST (seconds)</th>
<th>Speed Improve.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anderson*</td>
<td>1.0 M</td>
<td>1143 ([-0.01, 0.01])</td>
<td>792</td>
<td>7910</td>
<td>10.0</td>
</tr>
<tr>
<td>af_shell10</td>
<td>1.5 M</td>
<td>1045 ([2000, 2250])</td>
<td>37</td>
<td>274</td>
<td>7.4</td>
</tr>
</tbody>
</table>

*Anderson Model of Localization (Schrödinger eq.): \(H x = \lambda x\)\n\(\text{diag}(H): \text{random}; \text{offdiag}(H) = 1\)*
**Sampling: computing 400 eigenpairs**

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Size</th>
<th>Interval</th>
<th>$\text{abs. res. } 10^{-5}$</th>
<th>$\text{abs. res. } 10^{-9}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NASTRAN 1</strong></td>
<td>1.5M</td>
<td>$[-.01, 10^3]$</td>
<td>21 s</td>
<td>40 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>~.05 s/ep</td>
<td>~.10 s/ep</td>
</tr>
<tr>
<td><strong>NASTRAN 2</strong></td>
<td>7.2M</td>
<td>$[-.01, 3 \times 10^6]$</td>
<td>181 s</td>
<td>302 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>~.45 s/ep</td>
<td>~.76 s/ep</td>
</tr>
</tbody>
</table>

*Obtain the 4 eigenvalues closest to 100 points uniformly distributed in each interval and the corresponding eigenvectors using TraceMIN on 100 nodes*
Thank you!