Parallel Algorithms for Solution of Large Sparse Linear Systems with Applications

Murat Manguoğlu
Department of Computer Engineering
Middle East Technical University, Ankara, Turkey
Outline

- Gaussian elimination (LU factorization) vs parallel DS factorization
- Banded DS factorization
- Matrix reordering and banded preconditioning
- Sparse DS factorization
- Application:
  - OpenFOAM – 3D lid-driven cavity
  - FSI problem – arterial blood flow simulation
  - UF Sparse Matrix Collection
Target Computational Loop

Integration

Newton iteration

Linear system solvers

Most time consuming part in large simulations

$\eta_k$

$\epsilon_k$

$\Delta t$
Design principles

• Reducing memory references and interprocessor communications at the cost of extra arithmetic operations
• Allowing multiple levels of parallelism
• Creating many algorithms – versions vary depending on architectural characteristics of the underlying computing platform
DS factorization vs LU factorization

$A = \begin{bmatrix} 0 \\ \end{bmatrix}$

$L = \begin{bmatrix} 1 \\ \end{bmatrix}$

$U = \begin{bmatrix} 0 \\ \end{bmatrix}$

$A = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ \end{bmatrix}$

$D = \begin{bmatrix} D^{-1}A \\ \end{bmatrix}$

$S = \begin{bmatrix} ? \\ \end{bmatrix}$
Parallel solution of banded linear systems (SPIKE algorithm)

Solving $Ax=f$

**Step 1:** weighted matrix reordering (symmetric/nonsymmetric)

**Step 2:** extract a “banded” preconditioner, $M$, and solve the system:

Solve $Mz = r$ using DS fact. ($M$ is “banded”)

BiCGstab (or any other Krylov subspace method)

reordering with weights and extracting the preconditioner

Eigenvalues of the Laplacian of a Graph

• **Case 1:**
  – A is a symmetric matrix of order n
  – B = A
  – The weighted Laplacian matrix L is given by:
    • \( L(i,i) = \sum |B(i,k)| \quad \text{for} \quad k = 1,2,\ldots,n; \quad k \neq i \)
    • \( L(i,j) = -|B(i,j)| \quad \text{for} \quad i \neq j \)

• **Case 2:**
  – A is nonsymmetric
  – \( B = (|A| + |A^T|)/2 \)
  – L is obtained as in Case 1.
**The Fiedler vector**

- Obtain the eigenvector of the second smallest eigenvalue of:
  \[ L \mathbf{x} = \lambda \mathbf{x} \; ; \; \lambda := \{0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n\} \]

- Minimize \[ \sigma = \sum |B(i,j)| (x(i) - x(j))^2 \]
  for all \( i, j \) for which \( B(i,j) \neq 0 \), such that:
  \[ e^T \mathbf{x} = 0, \text{ and } \mathbf{x}^T \mathbf{x} = 1. \]

- In other words:
  \[ \sigma = \mathbf{x}^T L \mathbf{x} \]

---

**Trace Minimization (TraceMin)**

- $A \ x = \lambda \ B \ x$
  - $A := \text{symmetric}; \ B := \text{symmetric positive definite}$

- Minimize $\text{tr}(Y^T A Y)$ s.t. $(Y^T B Y) = I_p$

**Solution:**

$$\text{min tr}(Y^T A Y) = \sum \lambda_j \quad (j=1,2,\ldots,p)$$

$$\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \ldots \leq \lambda_p < \lambda_{p+1} \leq \ldots \leq \lambda_n$$

TraceMin iteration

Form a section

\[ Y^T_k A Y_k = \Sigma_k = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p)^{(k)} \]
\[ Y^T_k B Y_k = I_p \]

Solve Saddle-Pt Problem

\[
\begin{bmatrix}
    A & BY_k \\
    Y^T_k B & 0
\end{bmatrix}
\begin{bmatrix}
    Y_{k+1} \\
    L
\end{bmatrix}
= 
\begin{bmatrix}
    0 \\
    I_p
\end{bmatrix}
\]

Main Cost!!!
A Parallel Algorithm for Computing the Fiedler Vector: TraceMin-Fiedler

• Based on the Trace Minimization algorithm computes $\lambda_2$ and $v_2$ of $Lx = \lambda x$
  
  – Forms the small Schur complement explicitly
  – Solution of the linear systems involving $L$ via CG with diagonal preconditioning
  – $L$ is not spd (it is spsd)

• Comparison against the best sequential scheme (Harwell Subroutine MC73)


Source code(GPL): www.cs.purdue.edu/homes/mmanguog/fiedler.html
Data: $\mathbf{L}$ is the $n \times n$ Laplacian matrix defined in Eqn. (1), $\varepsilon_{out}$ is the stopping criterion for the $\|.\|_{\infty}$ of the eigenvalue problem residual.

Result: $x_2$ is the eigenvector corresponding to the second smallest eigenvalue of $\mathbf{L}$

$p \leftarrow 2; \quad q \leftarrow 3p$

$n_{conv} \leftarrow 0; \quad \mathbf{X}_{conv} \leftarrow [ ]$

$\hat{\mathbf{L}} \leftarrow \mathbf{L} + \|\mathbf{L}\|_{\infty} 10^{-12} \times \mathbf{I}$

$\mathbf{D} \leftarrow$ the diagonal of $\mathbf{L}$

$\hat{\mathbf{D}} \leftarrow$ the diagonal of $\hat{\mathbf{L}}$

$\mathbf{X}_1 \leftarrow \text{rand}(n,q)$

for $k = 1, 2, \ldots$, max_it do

1. Orthonormalize $\mathbf{X}_k$ into $\mathbf{V}_k$

2. Compute the interaction matrix $\mathbf{H}_k \leftarrow \mathbf{V}_k^T \mathbf{L} \mathbf{V}_k$

3. Compute the eigendecomposition $\mathbf{H}_k \mathbf{Y}_k = \mathbf{Y}_k \Sigma_k$ of $\mathbf{H}_k$. The eigenvalues $\Sigma_k$ are arranged in ascending order and the eigenvectors are chosen to be orthogonal.

4. Compute the corresponding Ritz vectors $\mathbf{X}_k \leftarrow \mathbf{V}_k \mathbf{Y}_k$

Note that $\mathbf{X}_k$ is a section, i.e. $\mathbf{X}_k^T \mathbf{L} \mathbf{X}_k = \Sigma_k, \mathbf{X}_k^T \mathbf{X}_k = \mathbf{I}$

5. Compute the relative residual $\|\mathbf{L} \mathbf{X}_k - \mathbf{X}_k \Sigma_k\|_{\infty} / \|\mathbf{L}\|_{\infty}$

6. Test for convergence: If the relative residual of an approximate eigenvector is less than $\varepsilon_{out}$, move that vector from $\mathbf{X}_k$ to $\mathbf{X}_{conv}$ and replace $n_{conv}$ by $n_{conv} + 1$ increment. If $n_{conv} \geq p$, stop.

7. Deflate: If $n_{conv} > 0$, $\mathbf{X}_k \leftarrow \mathbf{X}_k - \mathbf{X}_{conv}(\mathbf{X}_{conv}^T \mathbf{X}_k)$;

if $n_{conv} = 0$ then

Solve the linear system $\hat{\mathbf{L}} \mathbf{W}_k = \mathbf{X}_k$ approximately with relative residual $\varepsilon_{in}$ via the PCG scheme using the diagonal preconditioner $\hat{\mathbf{D}}$

else

Solve the linear system $\mathbf{L} \mathbf{W}_k = \mathbf{X}_k$ approximately with relative residual $\varepsilon_{in}$ via the PCG scheme using the diagonal preconditioner $\mathbf{D}$

9. Form the Schur complement $\mathbf{S}_k \leftarrow \mathbf{X}_k^T \mathbf{W}_k$

10. Solve the linear system $\mathbf{S}_k \mathbf{N}_k = \mathbf{X}_k^T \mathbf{X}_k$ for $\mathbf{N}_k$

11. Update $\mathbf{X}_{k+1} \leftarrow \mathbf{X}_k - \Delta_k = \mathbf{W}_k \mathbf{N}_k$
**f2: structural mechanics**

N: 71,505  NNZ: 5,294,285

Original matrix  
After MC73  
After TraceMin-Fiedler
Platform: a cluster where each node has two Intel (Westmere) X5670@2.93Ghz / 12 cores per node

Stopping tolerance: $\frac{||Lx - \lambda x||_2}{||L||_\infty} \leq 10^{-5}$
2\textsuperscript{nd} approach for solving $Ax=f$ 

Step 1: extract a “sparse” preconditioner, $M$, and solve the system: 

\begin{align*}
\text{Solve } Mz = r \text{ using DS factorization}
\end{align*}

BiCGstab
(or any other Krylov subspace method)

Parallel solution of sparse linear systems
(Domain Decomposing Parallel Solver - DDPS)
3D lid-driven cavity problem

- Standard test problem used in CFD community, problem becomes difficult as the Reynold's number increase.

- We use OpenFOAM to extract linear systems with 4 million unknowns and a Reynold's number of 5000.

- Curie cluster located at CEA, France is used. Curie consists of 360 “fat” nodes where each node has 4 eight core Intel EX X7560 processors running at 2.26 GHz and 128 GB of memory. The nodes are connected with an InfiniBand QDR Full Fat Tree network.
DDPS solver parameters:

- BiCGStab as the inner and outer iterative solver
- Incomplete-LU factorization (not threaded) with drop tolerance $10^{-2}$ and fill-in 5 for the diagonal blocks
- outer stopping tolerance is $10^{-10}$ and inner stopping tolerance is $10^{-2}$
Scalability (given as wall clock time in seconds for factor, solve and total) of DDPS using one MPI process per core on Curie.

<table>
<thead>
<tr>
<th>nodes</th>
<th>processes</th>
<th>cores</th>
<th>factor</th>
<th>solve</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>64</td>
<td>64</td>
<td>0,03</td>
<td>0,97</td>
<td>1,00</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>32</td>
<td>0,09</td>
<td>1,82</td>
<td>1,91</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>16</td>
<td>0,20</td>
<td>19,38</td>
<td>19,58</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>8</td>
<td>0,41</td>
<td>22,66</td>
<td>23,08</td>
</tr>
</tbody>
</table>
Scalability (given as wall clock time in seconds for factor, solve and total) of DDPS using 1 MPI process per node and 16 threads per process on Curie.

<table>
<thead>
<tr>
<th>nodes</th>
<th>processes</th>
<th>cores</th>
<th>factor</th>
<th>solve</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>64</td>
<td>1024</td>
<td>0,03</td>
<td>3,71</td>
<td>3,74</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>512</td>
<td>0,09</td>
<td>1,46</td>
<td>1,55</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>256</td>
<td>0,18</td>
<td>6,57</td>
<td>6,75</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>128</td>
<td>0,37</td>
<td>7,06</td>
<td>7,44</td>
</tr>
</tbody>
</table>
Size of the reduced system, number of outer iterations, average number of inner iterations and the final relative residual for DDPS solver using different number of partitions.

<table>
<thead>
<tr>
<th>part</th>
<th>red. sys. size</th>
<th>outer</th>
<th>avg. inner</th>
<th>rel. res</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>96</td>
<td>28</td>
<td>1,72</td>
<td>2,65E-11</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>26,5</td>
<td>1,83</td>
<td>1,27E-11</td>
</tr>
<tr>
<td>16</td>
<td>9600</td>
<td>24,5</td>
<td>1,19</td>
<td>2,57E-11</td>
</tr>
<tr>
<td>8</td>
<td>3200</td>
<td>24,5</td>
<td>1,16</td>
<td>3,47E-11</td>
</tr>
</tbody>
</table>
DDPS variations

DDPS-D: use the direct solver Pardiso for $D^{-1}A$

DDPS-I1: use ILUT($10^{-1}$, 5) for $D^{-1}A$

DDPS-I3: use ILUT($10^{-3}$, 5) for $D^{-1}A$

DDPS-I4: use ILUT($10^{-5}$, 10) for $D^{-1}A$

In all cases the outer stopping tolerance is $10^{-7}$ and inner stopping tolerance is $10^{-5}$
Flow field at time level 78 of the FSI computation. Stream ribbons are colored by the velocity magnitude, with the colors ranging from blue (low velocity) to red (high velocity).
Volumetric flow rate, with the triangle representing the instant (time level 78+0.5) when the test data is extracted.

Linear systems to solve at each nonlinear iteration has 1,399,566 unknowns and the coefficient matrix 167,638, 284 nonzeros.
cluster which consists of two Intel X5670 processors per node (total 12 cores per node) and 24 GB of ram.
<table>
<thead>
<tr>
<th>p</th>
<th>DDPS-D</th>
<th>DDPS-I1</th>
<th>DDPS-I2</th>
<th>DDPS-I3</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-</td>
<td>F</td>
<td>F</td>
<td>1001.6</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>F</td>
<td>F</td>
<td>562.6</td>
<td>F</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>F</td>
<td>F</td>
<td>344.6</td>
<td>F</td>
</tr>
<tr>
<td>16</td>
<td>397.1</td>
<td>F</td>
<td>F</td>
<td>227.3</td>
<td>F</td>
</tr>
<tr>
<td>32</td>
<td>164.6</td>
<td>F</td>
<td>F</td>
<td>106.3</td>
<td>F</td>
</tr>
<tr>
<td>64</td>
<td>80.9</td>
<td>F</td>
<td>F</td>
<td>61.3</td>
<td>F</td>
</tr>
<tr>
<td>128</td>
<td>40.0</td>
<td>F</td>
<td>F</td>
<td>37.0</td>
<td>F</td>
</tr>
<tr>
<td>256</td>
<td>33.0</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>$p$</td>
<td>DDPS-D</td>
<td>DDPS-I1</td>
<td>DDPS-I2</td>
<td>DDPS-I3</td>
<td>NONE</td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>F</td>
<td>F</td>
<td>1127.1</td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>F</td>
<td>F</td>
<td>582.7</td>
<td>F</td>
</tr>
<tr>
<td>8</td>
<td>-</td>
<td>F</td>
<td>F</td>
<td>408.7</td>
<td>F</td>
</tr>
<tr>
<td>16</td>
<td>398.2</td>
<td>F</td>
<td>F</td>
<td>228.1</td>
<td>F</td>
</tr>
<tr>
<td>32</td>
<td>178.1</td>
<td>F</td>
<td>F</td>
<td>113.3</td>
<td>F</td>
</tr>
<tr>
<td>64</td>
<td>76.3</td>
<td>F</td>
<td>F</td>
<td>63.8</td>
<td>F</td>
</tr>
<tr>
<td>128</td>
<td>45.4</td>
<td>F</td>
<td>F</td>
<td>35.6</td>
<td>F</td>
</tr>
<tr>
<td>256</td>
<td>23.1</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Robustness of DDPS-D  
- systems from UF sparse matrix collection-

<table>
<thead>
<tr>
<th>#</th>
<th>name</th>
<th>n</th>
<th>application</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>atmosmodl</td>
<td>1,489,752</td>
<td>computational fluid dynamics</td>
</tr>
<tr>
<td>2</td>
<td>hvdc2</td>
<td>189,860</td>
<td>power network analysis</td>
</tr>
<tr>
<td>3</td>
<td>language</td>
<td>399,130</td>
<td>directed weighted graph</td>
</tr>
<tr>
<td>4</td>
<td>ohne2</td>
<td>181,343</td>
<td>semiconductor device simulation</td>
</tr>
<tr>
<td>5</td>
<td>rajat31</td>
<td>4,690,002</td>
<td>circuit simulation</td>
</tr>
<tr>
<td>6</td>
<td>thermomech_dk</td>
<td>204,316</td>
<td>thermal simulation</td>
</tr>
<tr>
<td>7</td>
<td>tmt_unsym</td>
<td>917,825</td>
<td>electromagnetic simulation</td>
</tr>
<tr>
<td>8</td>
<td>torso3</td>
<td>259,156</td>
<td>2d/3d problem</td>
</tr>
<tr>
<td>9</td>
<td>xenon2</td>
<td>157,464</td>
<td>metarial science</td>
</tr>
</tbody>
</table>
Robustness of DDPS-D
- number of iterations -

<table>
<thead>
<tr>
<th>#</th>
<th>DDPS-D</th>
<th>ILUPACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.5</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>260</td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>5</td>
<td>106.5</td>
<td>F</td>
</tr>
<tr>
<td>6</td>
<td>752.5</td>
<td>31</td>
</tr>
<tr>
<td>7</td>
<td>212</td>
<td>F</td>
</tr>
<tr>
<td>8</td>
<td>8.5</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>67</td>
<td>F</td>
</tr>
</tbody>
</table>
Conclusions

- New factorization: DS vs. LU

- DDPS-D/DDPS-I solvers are scalable and robust involving inner-outer iterations and hence flexible

- Hybrid: combining direct and iterative solvers

- Hybrid: can use MPI/OpenMP at the same time
Acknowledgments

I would like to thank Ahmed Sameh, Ananth Grama, Eric Cox, Faisal Saied, Olaf Schenk, Madan Sathe, Joerg Hertzer, Kenji Takizawa, and Tayfun Tezduyar for the numerous discussions and for their support.

This work has been partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no: RI-261557 and METU BAP-08-11-2011-128 grant
Thank you!