Parallization of the LU Decompositon and the Conjugate Gradient Method

CS420 by Christopher Prinds Bilberg & Michel Robijns

Problem Statement

- Parallelization of methods for solving systems of linear equations
  - $A\mathbf{u} = \mathbf{f}$
  - Such systems commonly arise in scientific computing
  - Solving these systems is the most computationally intensive part of practically all numerical simulations

- A direct method: LU decomposition

- An iterative method: the conjugate gradient method

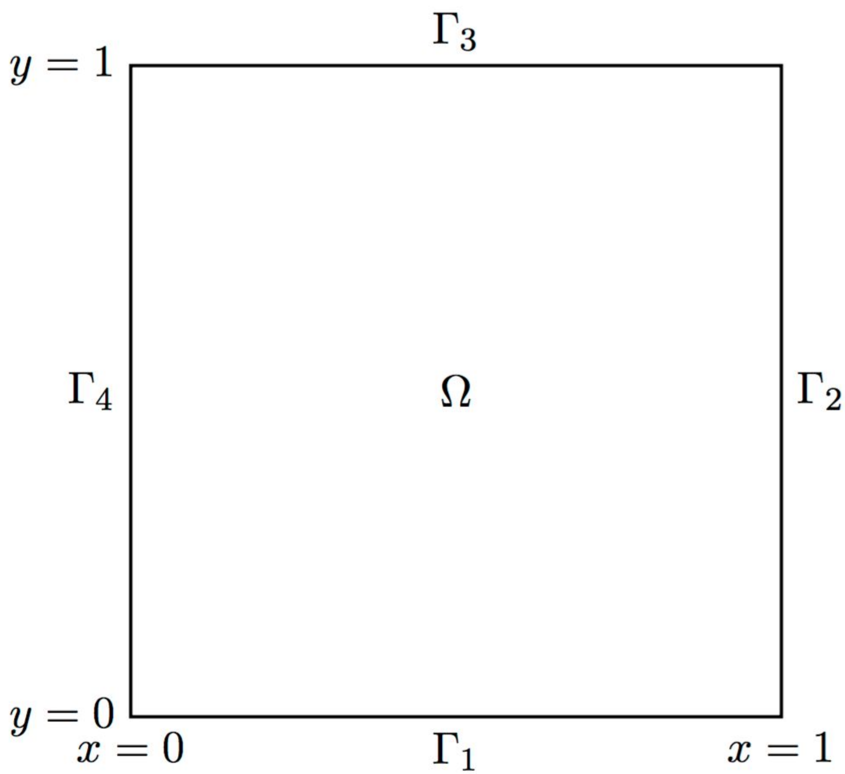
Sample Problem

- Finite-difference discretization of the Helmholtz equation
  - Has an exact solution
  - Symmetric and positive definite coefficient matrix

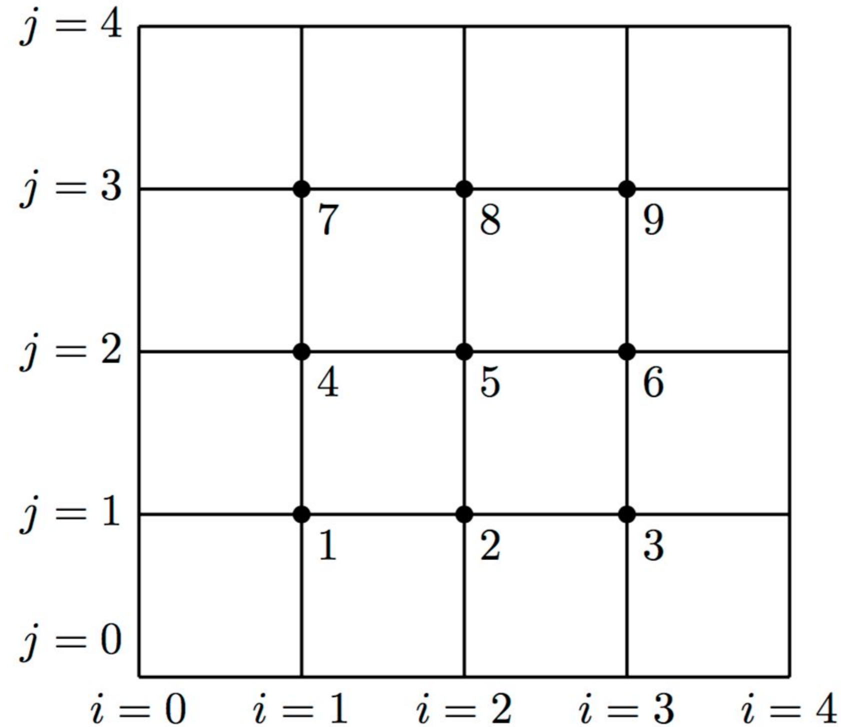
- $$-\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + 10u = f(x, y) \quad \text{on} \quad \Omega := (0,1) \times (0,1)$$
  $$u = g(x, y) \quad \text{on} \quad \partial\Omega := \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$$

- $u(x, y) = y \sin(xy)$

Sample Problem



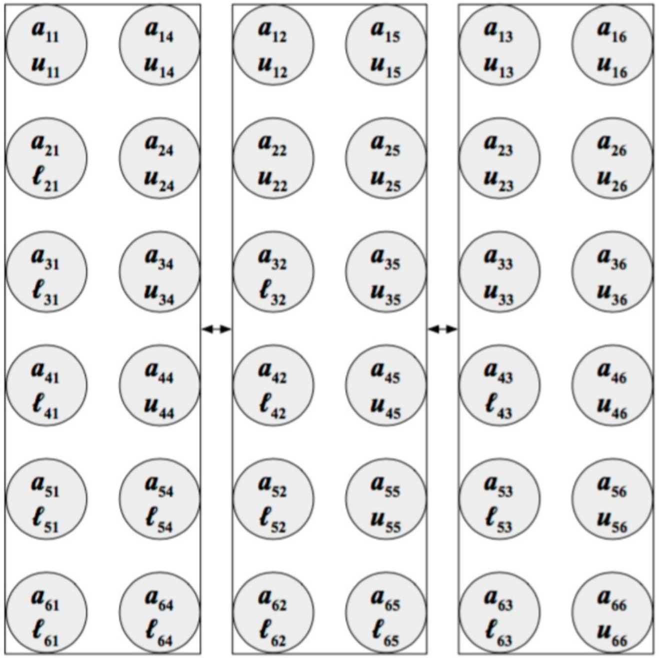
(a) Domain



(b) Discretization

LU Decomposition

- Direct method (no iterations)
- Based on Gaussian elimination
- $A\mathbf{u} = \mathbf{f} \Rightarrow LU\mathbf{u} = \mathbf{f}$
- Parallelization
  - Column-wise distribution in MPI
  - OpenMP for different 'for' loops

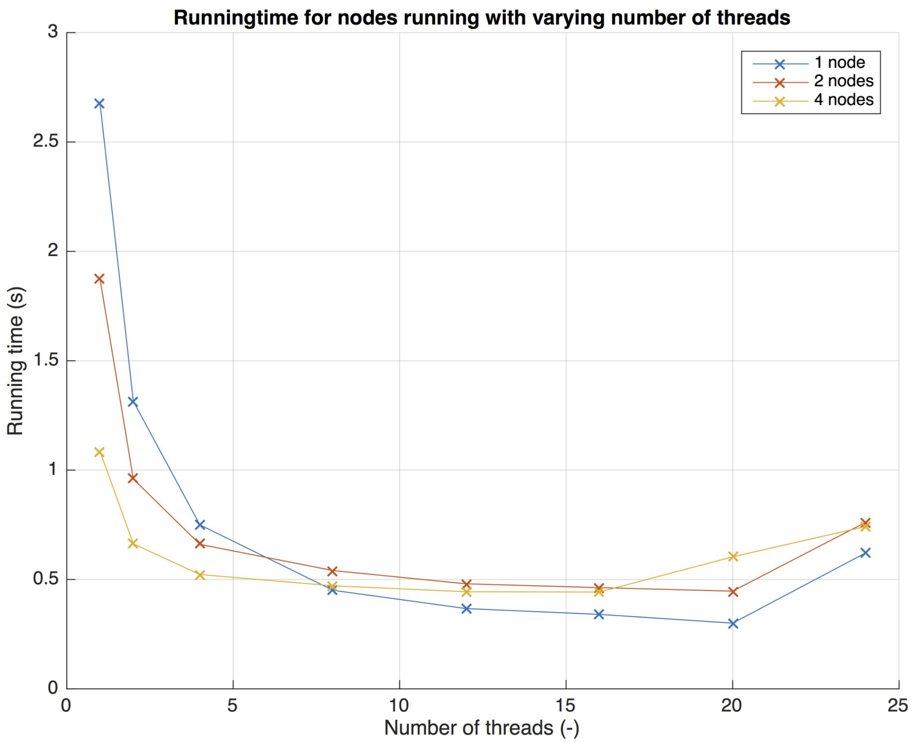


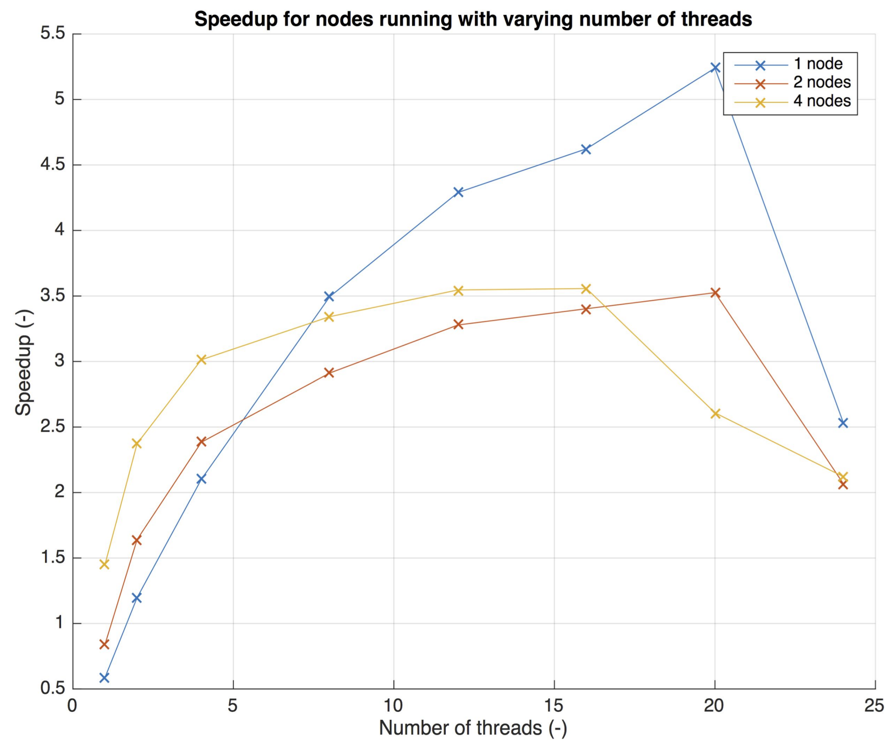
The Conjugate Gradient Method

- Iterative method
  - Making initial guess
  - Improving guess
- Parellization
  - MPI
    - Distributing matrix and vectors
  - OpenMP
    - Matrix-vector multiply
    - Inner product
    - a + k * b

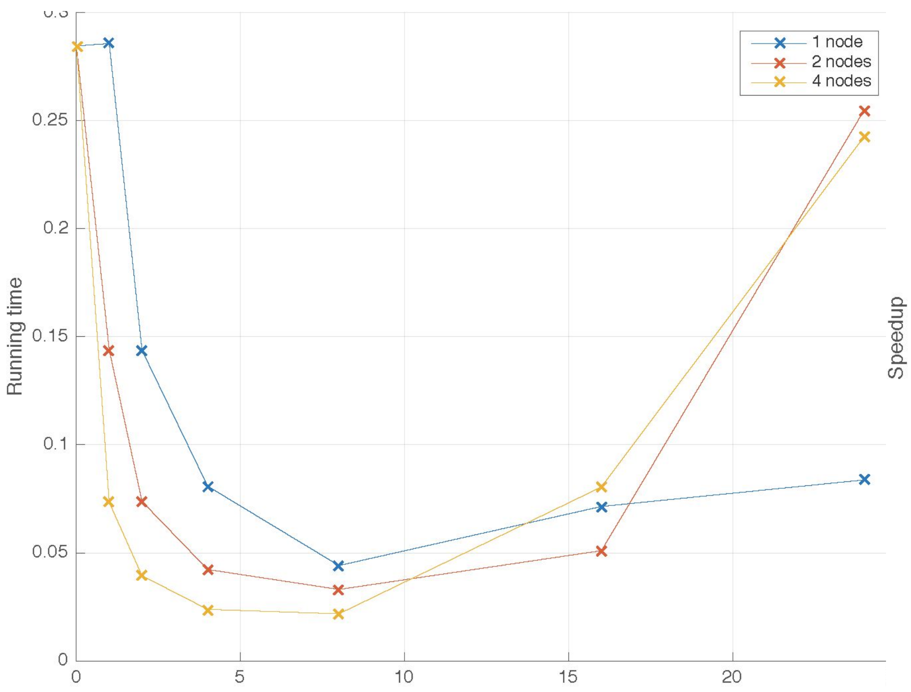
$k = 0; x_0 = 0, r_0 = b$
$\texttt{while}(||r_k||^2 > \text{tolerance})\ \texttt{and}\ (k < \texttt{max\_iter})$
$\quad k++$
$\quad \texttt{if}\ k = 1$
$\qquad p_1 = r_0$
$\quad \texttt{else}$
$\qquad \beta = \frac{r_{k-1} \cdot r_{k-1}}{r_{k-2} \cdot r_{k-2}}$
$\qquad p_k = r_{k-1} + \beta_k \cdot p_{k-1}$
$\quad \texttt{endif}$
$\quad s_k = Ap_k$
$\quad \alpha_k = \frac{r_{k-1} \cdot r_{k-1}}{p_k \cdot s_k}$
$\quad x_k = x_{k-1} + \alpha_k \cdot p_k$
$\quad r_k = r_{k-1} - \alpha_k \cdot s_k$
$\texttt{endwhile}$
$x = x\_k$

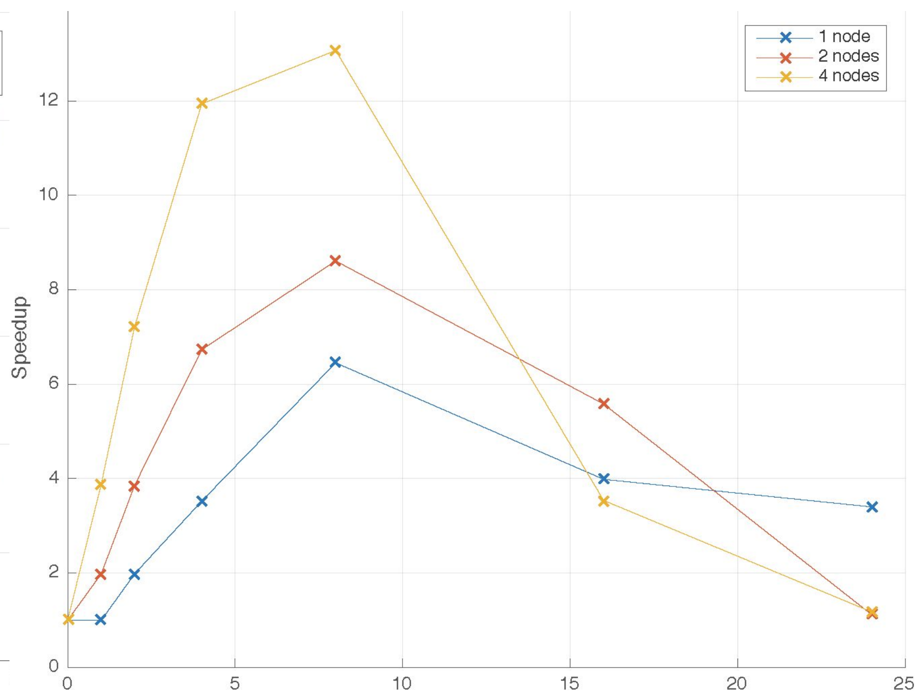
Results - LU Decomposition





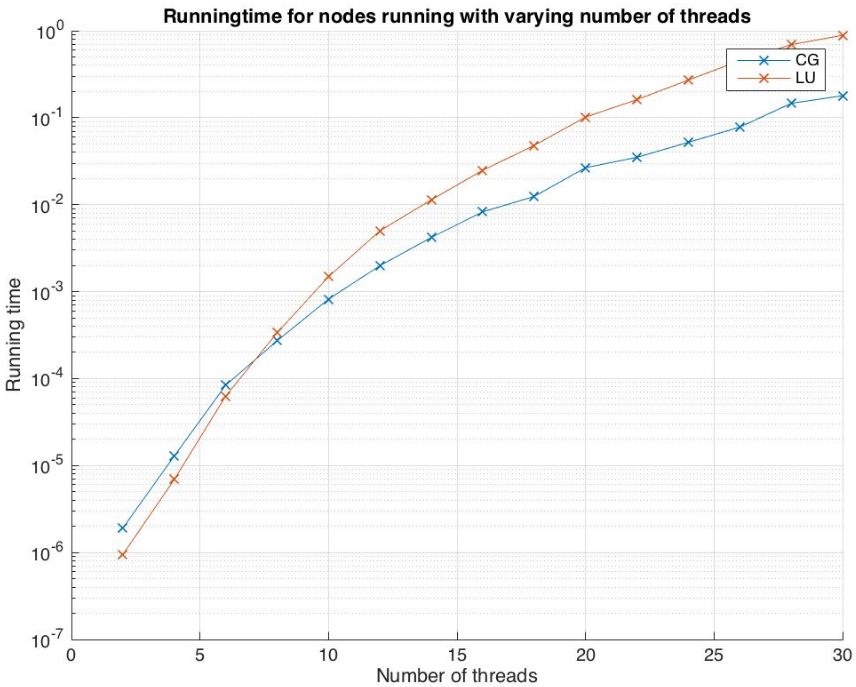
Results - Conjugate Gradient Method





Comparison

- LU best for smaller problem size
- CG faster for bigger problem size



Conclusion

- LU decomposition is faster for small problems
- The conjugate gradient method is faster for large problems
- LU decomposition is hard to parallelize on distributed memory systems
- The conjugate gradient method is suitable for distributed memory systems